BACHELOR

Fair assignment in social choice theory with an application to the assignment of students to high-schools

Struijs, Martijn

Award date:
2016

Fair assignment in social choice theory with an application to the assignment of students to high-schools

EINDHOVEN UNIVERSITY OF TECHNOLOGY
Author: Martijn Struijs
Supervisor: Gerhard J. Woeginger

June 29, 2016

1 Introduction

Since 2005, the city of Amsterdam uses a centralised system to assign 8000 students to 700 high-schools in the entire city [9]. Before this system, most students applied at only a few popular schools for their first choice, which means a large number of students got rejected due to capacity constraints and had to apply to other schools. In this new system, all students hand in a list of their preferences for the schools to OSVO, the association of high-schools in Amsterdam, which determines the assignment of all students to the high-schools with an algorithm. The first system was mainly an administrative improvement. However, in 2015, another assignment algorithm was used, with the goal of creating a fair assignment for the students. This algorithm is a variant of the Deferred Acceptance (DA) algorithm, which creates a stable matching for the preferences of both the schools and the students [5]. The results of these algorithms have been studied in practice and in a simulation study [7, 8].

An important difference of the situation in Amsterdam with most school assignment problems in the literature is that the schools do not actually express preferences over students. This means that a DA variant needs to assign artificial preference to the schools, which appears inefficient. Hence, most theoretical research on the case where only the students have preferences focusses on other methods, such as the Serial Dictatorship (SD) algorithm. Another aspect of this situation is that all students should be assigned to some school and it is allowed that some schools have no students assigned to them.

The goal of this thesis is to explore the theoretical possibilities of this school assignment problem with one-sided preferences and attempt to create an improvement for the algorithm currently used in Amsterdam.

The structure of the paper is as follows: First, we define our model of this situation and provide some basic definitions. Next, we study the SD algorithm more closely, consider some variants and describe their properties. After that, we investigate the theoretical properties of *anonymity*. In particular, we show that no matching mechanism can be both anonymous and *strategy proof*. We discuss the applicability of the Serial Dictatorship with Project Closures (SPDC) [13], for the case with minimum quorums. We present the Beneficial Serial Dictatorship (BSD) algorithm, which is both strategy proof and *pareto efficient*, as is SPDC, but also will always give a feasible assignment that leaves no students unassigned if such an assignment exists, unlike SPDC. Finally, we look at present a conclusion and give some general recommendations.

There is plenty of literature on the case with 2-sided preferences. Most of the 2-sided matching algorithms are based on the Deferred Acceptance (DA) algorithm by Gale and Shapely [5]. The problem with 1-sided preferences can be considered as a special case if the former if we construct some random 'artificial' (i.e. created by the algorithm) preferences for the schools. There are roughly two ways to do this. DA with Multiple Tie-Breaking (DA-MTB) achieves this by randomly assigning preferences to all schools. DA with Single Tie-Breaking (DA-STB) does this by assigning a single, randomly chosen, preference to all schools.

Fragiadakis et al. [4] consider several algorithms for the 2-sided problem with minimum quorums. Additionally, they give an extension of SD to use minimum quorums in the 1-sided problem. This extension is less complex than the BSD algorithm proposed in this paper, but it solves a slightly different problem. The extension by Fragiadakis et al. requires that all schools in H have at least some student assigned to them, so they must meet their minimum quorum, whereas our algorithm does not have such a restriction, giving more options for the students.

For more information on the specific situation in Amsterdam and the Dutch secondary education system in general, we refer to [9].

2 Model

Throughout this paper, we consider 3 different assignment scenarios. Two of them are special cases of the third one, so our model describes the latter. In particular, our model concerns itself with the case in which schools have both a maximum capacity and a minimum quorum, as in [13], although our model is slightly different.

Let $S = \{s_1, \dots, s_n\}$, be the set of students and $H = \{h_1, \dots, h_m\}$ the set of high-schools, for $m, n \in \mathbb{N}$. Each school $h \in H$ has a maximum capacity $c_h \in \mathbb{N}_+$ and a minimum quorum $q_h \in \mathbb{N}_+$. Furthermore, $q_h \leq c_h$. This means that every school must have at least q_h and at most c_h students assigned to it, or no students assigned at all. So, for simplicity, we assume that $q_h \leq n$, since no student could be assigned to such a school otherwise. We write $c = (c_h)_{h \in H}$ and $q = (q_h)_{h \in H}$.

Every student $s \in S$ has a preference ordering $\succeq_s$ over the schools and the empty set $\emptyset$, which represents not being assigned to any school at all. $\succ_s$ is the associated strict preference relation. The total preference profile $(\succeq_s)_{s \in S}$ is denoted by $\succeq$. We assume two things about these preferences:

(1) Each student's preference ordering $\succeq_s$ is strict, i.e. $h \succeq_s h'$ if either $h \succ_s h'$ or $h = h'$; and

(2) $h \succ_s \emptyset$, for all $s \in S$ and $h \in H$; i.e. every student prefers to be assigned to some school, to not being assigned at all.

The rank $R_{\succeq, s}(h)$ is 1 plus the number of schools in H that student s prefers over h, under the preference $\succeq$.

A matching μ is a mapping $\mu : S \cup H \to S \cup H$, such that

(i) $\mu(s) \subseteq H$, for all $s \in S$;

(ii) $\mu(h) \subseteq S$, for all $h \in H$; and

(iii) $h \in \mu(s)$ if and only if $s \in \mu(h)$.

If $\mu(s) = \emptyset$, we say that s is unmatched or unassigned in μ or that s is not assigned to any school in μ. Similarly, if $\mu(h) = \emptyset$, we say that h is unmatched in μ. For convenience, we will write $\mu(s) = h$ instead of $\mu(s) = \{h\}$. We will focus on matchings that respect the quorums and capacities of the schools.

Definition 1 (Feasible Matching). A matching μ is feasible if the following two conditions hold:

(i) for all $h \in H$, either $q_h \leq |\mu(h)| \leq c_h$ or $\mu(h) = \emptyset$; and

(ii) $|\mu(s)| = 1$, for all $s \in S$.

The set of all feasible matchings is denoted by $\mathcal{M}$.

Note that in the previous definition, condition (ii) is more strict than in the definition of feasibility by Monte and Tumennasan [13], who allow students to be unassigned. The definition of Pareto efficiency in this model is standard.

Definition 2 (Pareto efficiency)**.** A *feasible* matching $\bar{\mu}$ *Pareto dominates* a feasible matching μ if

(i) $\bar{\mu}(s) \succ_s \mu(s)$ for at least one $s \in S$; and

(ii) $\bar{\mu}(t) \succeq_t \mu(t)$ for all $t \in S$.

A matching μ is *Pareto efficient* if it is feasible and, in addition, there does not exist any feasible matching $\bar{\mu}$ that Pareto dominates μ.

An assignment problem P with quorums is given by the set of students, schools, the quorums and capacities of these schools and the students' preference profile, i.e.: $P = (S, H, q, c, \succeq)$. An assignment problem P is feasible if there exists a feasible matching for P.

A mechanism φ for such a problem is a function that assigns a matching for each problem. A mechanism is Pareto efficient if it results in a Pareto efficient matching for all feasible assignment problems. Similarly, a mechanism is feasible if it produces a feasible matching for all feasible assignment problems.

Some mechanisms iterate over a number of steps and create a temporary matching of the students and schools in each step. $\tilde{\mu}_t$ is the temporary matching at step t. The remaining capacity $c_h(t)$ of a school h after step t equals $c_h - |\tilde{\mu}_t(h)|$ and the remaining quorum $q_h(t)$ of school h after step t equals $\max\{q_h - |\tilde{\mu}_t(h)|, 0\}$.

Definition 3 (Strategy Proofness)**.** Let φ be a mechanism for the set of assignment problems with quorums. Call φ (individually) *manipulable* if there exist two problems $P = (S, H, q, c, \succeq)$ and $P' = (S, H, q, c, \succeq')$ and some $s \in S$ such that:

(i) $\succeq'$ is an *alteration* of $\succeq$ by student s: $\succeq'$ differs from $\succeq$ only in the preference ordering of student s (i.e. $\succeq_s \neq \succeq'_s$ and $\succeq_t = \succeq'_t$ for all $t \neq s$); and

(ii) $\mu'(s) \succ_s \mu(s)$, where $\mu := \varphi(P)$ and $\mu' := \varphi(P')$.

A mechanism φ is *strategy proof* if it is not manipulable by any student $s \in S$.

Although these properties are interesting to investigate for certain mechanisms, we lack an intuitive representation of the solutions provided by these algorithms. A useful method is the so-called *envy graph* [2].

Definition 4 (Envy Graph)**.** Given a fixed assignment problem P, an *envy graph* $G(\mu)$ of a feasible matching μ under P is a directed graph with all students S as vertex set. There is an edge from student s to student t if and only if $\mu(t) \succ_s \mu(s)$, i.e. if student s prefers the assignment of student t over his own assignment.

A matching μ is Pareto optimal, if and only $G(\mu)$ is cycle-free [2].

3 Serial dictatorship and related mechanisms

There are many possible mechanisms that satisfy various properties. Most of these mechanism belong to a parametrised family of mechanisms. If they are of practical use, these families are often called "algorithms". This name is chosen, since most algorithms that solve the multiple assignment problem non-deterministically choose a member of such a family, usually for fairness purposes, as most mechanisms favour certain students or schools. So, as we intend to study deterministic mechanisms, we will usually describe these mechanism classes instead of their related non-deterministic algorithms.

Throughout this section, we analyse problems without an effective minimum quorum, so for all $h \in H : q_h = 1$. In this section, we therefore write an assignment problem as $P = (S, H, \succeq, c)$, unless stated otherwise.

The most extensively studied algorithm is known as the serial dictatorship algorithm (SD) [1].

Algorithm 1 (Serial dictatorship)**.** Given an assignment problem P and an enumeration E of S, the *Serial Dictatorship* algorithm does the following for every student $s \in S$, in the order of E: Assign s to the most preferred school of s, of all schools that have not reached their maximum capacity yet.

Remark. The SD algorithm takes exactly n steps, in which it needs to check for all schools whether it has remaining capacity for the worst case. So, the computational complexity of this algorithm is $\mathcal{O}(mn)$.

Note that the SD algorithm gives the same results as DA-STB. In DA-STB, every student eventually gets assigned to the most preferred school that is available after all students with an earlier place in the single 'lottery' have been assigned, which SD does directly. SD is both simple and has several good properties. However, for a fixed enumeration, it seems rather unfair. In practice, such an enumeration is therefore often chosen from a uniform random distribution. This is called the Random Serial Dictatorship algorithm (RSD). As this algorithm ignores minimum quorums, it is not suitable for that case, which we will show in Example 3.

Before we present some new results, we will first show some well known properties regarding some of the Serial Dictatorship mechanism.

Theorem 1 (Properties of Serial Dictatorship)**.** *Serial Dictatorship is a Pareto efficient, strategy-proof and resource monotonic mechanism.*

Proof. For the following, let $P = (S, H, \succeq)$ be a feasible assignment problem and let E an enumeration of S. SD is the Serial Dictatorship mechanism with the enumeration E. Matching μ equals SD(P).

Pareto efficiency: Note that the matching μ is feasible, since the SD mechanisms assigns every student to some school in every step if any school has remaining capacity. Since P is feasible, some school has remaining capacity at every step, which means μ is feasible.

Toward a contradiction, suppose that there exists some matching $\bar{\mu}$ that Pareto dominates μ. This means there exist $s \in S$ such that $\bar{\mu}(s) \succ_s \mu(s)$. s^* is the first such student in the order given by E, and define $h^* := \bar{\mu}(s^*)$. Since s^* prefers h^* over $\mu(s^*)$, there was no remaining capacity at school h^* when the SD algorithm reached s^*. This means that, since s^* is assigned to h^* in $\bar{\mu}$, there is a student $s' \neq s^*$ with $\mu(s') = h^*$ and $\bar{\mu}(s') \neq h^*$. Since $\bar{\mu}$ Pareto dominates μ, we have that $\bar{\mu}(s') \succ_{s'} \mu(s')$. However, since $\mu(s') = h^*$, we have that s' occurs before s^* in E, which contradicts s^* being the first such student. Hence, there exists no matching that Pareto dominates μ and μ is feasible, so μ is Pareto efficient.

Strategy proofness: Let $P' = (S, H, \succeq')$, with $\succeq'$ an alteration of $\succeq$ by some student s. Define $\mu' :=$ SD(P'). Since all students other than s have the same preference as in P, SD assigns the same school to all students assigned before s. So, the set of available schools for s is the same in both cases. Since SD assigns to s the best of these schools, according to the given preference, we have that $\mu(s) \succeq_s \mu'(s)$, since for P, SD assigns the optimal school for s in this set under $\succeq$. So, $\succeq'$ cannot be a manipulation and since $\succeq'$ is arbitrary, no manipulation is possible, which means SD is strategy proof. □

Another interesting property is that any Pareto efficient matching can be generated by the SD algorithm [1]. A simpler proof is provided by Abraham et al. [2], who show that for any

Pareto efficient matching μ, using the reverse topological sort of the envy graph $G(\mu)$ as the enumeration for the SD algorithm results in μ. This result does not imply that SD is the 'perfect' mechanism, since we can satisfy even better properties if we restrict the resulting matchings to a subset of the Pareto efficient matchings. The Rank minimizing mechanism, a special case of the linear programming method as described in [3], is such a mechanism.

Definition 5 (Rank minimizing mechanism)**.** Let E be an enumeration of S. The total rank cost $C(\mu)$ of a matching μ equals $\sum_{s \in S} R_{\succeq,s}(\mu(s))$. The set of matchings M equals $\{\mu \in \mathcal{M} |\ C(\mu) = \min_{\mu \in \mathcal{M}} C(\mu)\}$.

Define $M'(s)$, the most preferred subset of M by student s, as equal to $\{\mu \in M |\ \forall \mu' \in M : \mu(s) \succeq_s \mu'(s)\}$. Enumerate over all students s in the order of E. If $|M| \neq 1$, assign $M := M'(s)$. Otherwise, halt the iteration and return the only matching in M as the result of this mechanism.

Define rankMin as the Rank minimizing mechanism, for the given enumeration E.

Note that M will eventually contain exactly one matching. For two distinct matchings μ and μ', exists a student s, such that $\mu(s) \succ_s \mu'(s)$, as the preferences of the students are strict. So, if we enumerate over all students, we will eventually 'filter out' all but one matching.

The Rank minimizing mechanism is Pareto efficient. Indeed, if a matching μ is Pareto dominated by the matching $\bar{\mu}$, then at least one of the individual ranks is strictly smaller, while none have increased. So, the sum of all ranks of $\bar{\mu}$ is strictly smaller than the sum of μ. This means any matching where the sum of the ranks of that matching attains the minimum must be Pareto efficient.

However, the Rank minimizing mechanism is not strategy proof. An example is the following manipulation:

Example 1. Let $S = \{a, b, c, d\}$ and $H = \{1, 2, 3, 4\}$. The preferences are defined as follows:

$$1 \succ_a 2 \succ_a 3 \succ_a 4,$$
$$1 \succ_b 2 \succ_b 3 \succ_b 4,$$
$$1 \succ_c 3 \succ_c 2 \succ_c 4,$$
$$1 \succ_d 3 \succ_d 4 \succ_d 2$$

For this problem, the Rank minimizing mechanism must assign student d to school 4 to get a total cost of 8, which is the minimum. Consider the alteration $\succeq'$ by student d, with $3 \succ'_d 1 \succ'_d 2 \succ'_d 4$. Now, the new minimum total cost of 8 is attained only if d is assigned to school 3. Since $3 \succ_d 4$, this a manipulation by student d. Note that this an example where the success of a manipulation is independent of the enumeration used by rankMin. In fact, this example is minimal in the number of students for this property.

This manipulation appears complicated, even on a small problem. Even though this mechanism is not strategy proof, it might be a good mechanism if constructing an (individual) manipulation is a sufficiently 'hard' problem.

One way to formalise this notion is to consider the computational complexity of this (algorithmic) problem. A problem is NP-hard if a certain known difficult problem it can be reduced to our original problem in polynomial time. It is widely believed that these problems cannot be solved in polynomial time. [14] Therefore, if we can show that manipulation of the Rank minimizing mechanism is NP-hard, then even though manipulation is possible, it can be too hard in practice. Unfortunately, this is not the case, and we will provide an algorithm can produce a manipulation or determine that no manipulation is possible, in polynomial time. We first define our sub-problem.

Definition 6 (Reduced problem)**.** The *reduced (assignment) problem* $P_{s,h}$ of an assignment problem P for a student $s \in S$ and a school $h \in H$ is defined, such that a matching μ is feasible for $P_{s,h}$ if μ is feasible for P and $\mu(s) = h$. Define rankMin$(P_{s,h})$, by using Definition 5, but with $M = \{\mu \in \mathcal{M}' \mid C(\mu) = \min_{\mu \in \mathcal{M}'} C(\mu)\}$, where $\mathcal{M}' = \{\mu \in \mathcal{M} \mid \mu(s) = h\}$.

Note that the reduced problem $P_{s,h}$ differs from the problem P', with $P' = (S - s, H - h, \succeq)$. For P', the preference ordering $\succeq$ is defined on $H - h$, which means the rankings of the students differ from P. For the reduced problem, the rankings are the same as in P.

To simplify our algorithm, we will restrict ourselves to finding alterations that are effective for all enumerations, as in Example 1.

Instead of finding a manipulation, the following algorithm solves a slightly more general problem: it gives an alteration $\succeq'$ for student s^* such that the Rank minimizing mechanism assigns student s^* to some target school h^* or state that no such alteration exists. Clearly, we can find a manipulation by applying this algorithm to all schools.

Algorithm 2. Let P be an assignment problem, with $P = (S, H, \succeq)$ and let E be an enumeration of the students S. Furthermore, a student s^* and a target school h^* are given.

The algorithm is as follows: First, compute the minimum cost $C(h)$ of the reduced problem $P_{s^*,h}$, so $C(h) = C(\text{rankMin}(P_{s^*,h}, E))$, for all enumerations E. Next, store all $h \in H - h^*$ in the array $A[1..n-1]$ and sort A decreasing in $C(h)$. Construct the preference list R, such that $R[h^*] = 1$ and $R[A[i]] = i + 1$, for all $i \in \{1, \dots, n-1\}$. Construct the alteration $\succeq'$ of $\succeq$ by s^*, such that R is the preference list of h^* for alteration $\succeq'$. If it holds that $C(h^*) + R(h^*) < \min_{h \in H - h^*} C(h) + R(h)$, return $\succeq'$. Otherwise, report that the requested alteration does not exist.

Theorem 2. *Let P an assignment problem, with $P = (S, H, \succeq)$, let s^* a student in S and let h^* a school in H. Algorithm 2 constructs a alteration $\succeq'$ of $\succeq$ by s^* such that, for all enumerations E of S,* $\text{rankMin}(P, E)$ *assigns s^* to h^* or Algorithm 2 correctly reports that no such alteration is possible, in polynomial time.*

Proof. Correctness:

For a school h and ranking R, write $C_R(h)$ to denote the minimal cost for rankMin to assign student s^* to school h under ranking R, so $C_R(h) = C_h + R(h)$.

First, we claim that for a ranking R, s^* gets assigned to h^* for all enumerations E if and only if

$$\min_{h \in H - h^*} C_R(h) - C_R(h^*) > 0. \tag{1}$$

If Equation 1 holds, then only matchings μ with $\mu(s^*) = h^*$ have a minimum ranking cost, so rankMin must return such a matching. If Equation 1 does not hold, then there is a matching μ' with $\mu'(s^*) \neq h^*$ and minimal cost. Since μ' is Pareto efficient, there is at least one student s that strictly prefers this matching over all matchings μ with $\mu(s^*) = h^*$. Therefore, for an enumeration where s is first, s^* will not be assigned to h^*. Therefore, our claim holds.

R' is the ranking of student s^* under the preference $\succeq'$, as given by Algorithm 2. We will show that this reaches the maximum value for the left side of Equation 1 of all possible rankings for s^*. Suppose there exists a ranking R^* that satisfies (1). If $R^* = R'$, then we are done. Otherwise, we have that $R^*(h^*) \neq R(h^*)$ or there is at least one index $i \in \mathbb{N}$, such that $R^*(A[i]) \neq R(A[i])$. We will show that is possible to modify R^* into R' such that there is at least one school h less for which $R'(h) \neq R(h)$ (compared with R^*), while maintaining that R' satisfies (1).

If $R^*(h^*) \neq R(h^*)$, then $R^*(h^*) = k$, for some $k > 1$. Modify R^* into R' such that $R'(h^*) = 1$ and $R'(h) = k$, where h is such that $R^*(h) = 1$. For all other schools, R' and R^* are identical. Since our modification did not decrease the ranking for any school in $H - h^*$, we have that $\min_{h \in H - h^*} C_{R'}(h) \geq \min_{h \in H - h^*} C_{R^*}(h)$. Furthermore, $R^*(h^*) > R'(h^*)$, so

$$\min_{h \in H - h^*} C_{R'}(h) - C_{R'}(h^*) \geq \min_{h \in H - h^*} C_{R^*}(h) - C_{R^*}(h^*) > 0. \tag{2}$$

If there is an index $i \in \mathbb{N}$ such that $R^*(A[i]) \neq R(A[i])$, then define i^* as the smallest such index. Figure 1 gives an illustration of this case. The integer k is the rank of $R^*(A[i^*])$. Since

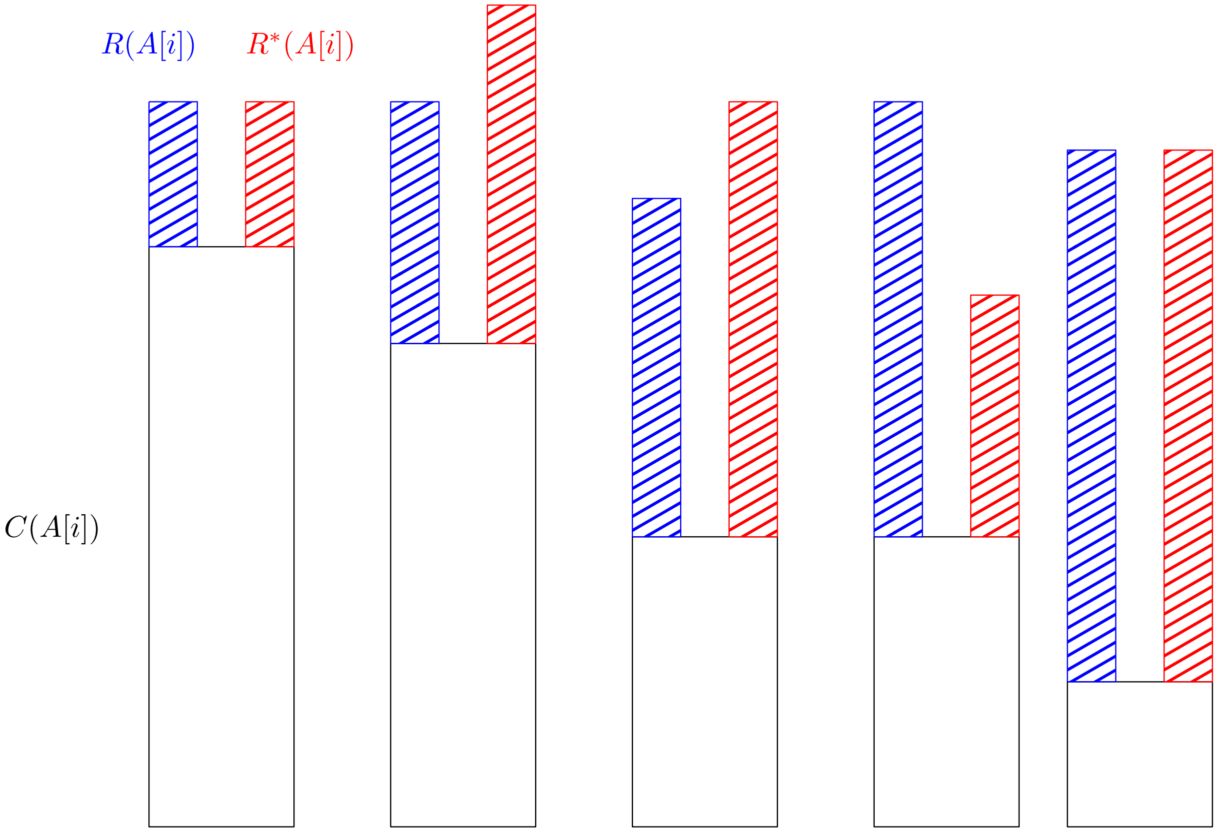

Figure 1: A visual representation of the costs for different indices i, as used in the proof of Theorem 2, where the length of the bars represent the size of the number. The bars on the bottom represents the value of the sub-problem. The left and right bar on top represent the added cost with ranking R and R^*, respectively. Observe that R maximizes the minimum summed cost.

$R^*[A[i]] = R[A[i]]$ for all $i < i^*$ and all those ranks are smaller than $R^*(A[i^*])$, it follows that $k > i^* + 1$. The index j^* is such that $R^*(A[j^*]) = i^* + 1$. Since $R^*(A[j^*]) \neq R(A[j^*])$, we have that $j^* > i^*$, as i^* is the smallest index with that property. Modify R^* into R', such that $R'(A[i^*]) = i^* + 1$, $R'(A[j^*]) = k$ and $R'(h) = R^*(h)$ for all $h \in H - \{A[i^*], A[j^*]\}$. We have that $R^*(A[i^*]) = k > i^* + 1 = R^*(A[j^*])$, so $C_{R^*}(A[j^*]) = \min_{i \in \{i^*, j^*\}} C_{R^*}(A[i])$. Furthermore, we have $R'(A[j^*]) = k > i^* + 1 = R^*(A[j^*])$ and $R'(A[i^*]) + C(A[i^*]) \geq R'(A[i^*]) + C(A[j^*]) = R^*(A[j^*]) + C(A[j^*])$. This means that

$$\min_{i \in \{i^*, j^*\}} C_{R'}(A[i]) \geq \min_{i \in \{i^*, j^*\}} C_{R^*}(A[i]).$$

All other schools remain unchanged, so (2) holds in this case as well.

In both cases, we construct a modification R' that differs less from R and satisfies (1), so by induction, we have that R satisfies (1). Therefore, Algorithm 2 gives correct output in this case.

If there is no ranking that satisfies (1), then R does not satisfy (1), so Algorithm 2 gives correct output.

Running time: Define $n = \max(|S|, |H|)$. Computing $C(h)$ can be done by solving the minimum weighted bipartite matching problem, for which the 'Hungarian method' [12] achieves a running time of $\mathcal{O}(n^3)$. This is done at most n times, so this takes $\mathcal{O}(n^4)$ time. Next, sorting A takes $\mathcal{O}(n \log n)$ time. So, Algorithm 2 takes $\mathcal{O}(n^4)$ time. □

The problem of finding a manipulation can be solved by running the Algorithm 2 for all schools that are preferred to the current school of s^*, so this takes at most $\mathcal{O}(n^5)$ time, which is polynomial.

Even though this problem is not hard in the computational sense, this does not mean this problem is not hard in the information-theoretical sense. For a manipulation, we assume that the manipulating student is aware of the preferences of all other students. We handily made use of this in Algorithm 2, but this does not seem realistic in practice. A student is probably only aware of his own preference, the preference of some other students (e.g. his friends) and the general popularity of the schools (e.g. the likelihood some student will rank a certain school high on his list). While we suspect that manipulation of the Rank minimizing mechanism is sufficiently hard or even impossible under these constraints, we have not been able to formalise or prove this.

It might be useful to consider a weaker form of Pareto efficiency, as this could include more mechanisms.

Definition 7 (k-coalition stable)**.** Given a fixed assignment problem P, a matching μ is considered k-coalition stable if all cycles in the envy graph $G(\mu)$ have at least $k+1$ students.

Intuitively, this means no 'coalition' of at most k students can, by trading with their assigned positions within this coalition, reach a strictly more favourable matching for all members of the coalition.

Note that in the previous definition, if $|S| \leq k$, no cycles in the graph are allowed at all, so our statement is equivalent with Pareto efficiency. Therefore, Pareto efficiency implies k-coalition stability, so it can indeed be considered a weaker form of Pareto efficiency.

In fact, in the choice for the first system for the assignment problem in Amsterdam, the school association was fine with the fact that the system was not Pareto efficient, since there would be fewer students assigned to very low ranking schools, in practice. The problem was that, for the mechanism to be strategy proof, trading between the assigned schools of the students was not allowed, even if both students would be assigned to a school of better preference. Therefore, requiring k-coalition stability for some large enough k, such that the probability of that coalitions are being discovered and used is insignificant, will be acceptable as a replacement for Pareto efficiency.

However, if we additionally require our mechanism to be strategy proof, there will be assignment problems for which the mechanism must be Pareto efficient regardless:

Lemma 1. *For all $k \geq 2$: if a mechanism φ is strategy proof and k-coalition stable, then there exists an assignment problem where an k-coalition stable matching exists, but for which the matching given by φ is Pareto efficient.*

Proof. For $n = k+1$, consider the assignment problem $P = (S, H, \succeq)$, with the students $S = \{s_1, \ldots, s_n\}$ and schools $H = \{h_1, \ldots, h_n\}$. For all $i \in \{1, \ldots, n\}$, define $m := (i+1) \bmod n$ and define the preferences $\succeq$ such that $h_m \succ_{s_i} h_i$ and for all $j \in H - \{h_i, h_m\} : h_i \succ_{s_i} h_j$. For a matching μ that is both k-coalition stable and not Pareto efficient, there must be at least one cycle of n students in $G(\mu)$ and no cycle with fewer students. Since there are exactly n students, there must exist exactly one cycle of length n in $G(\mu)$ and no other edge, since any other edge would create a smaller cycle, given the cycle of length n. So, given the previous constraints and preferences, the only feasible matching for in P that is k-coalition proof but not Pareto efficient is the matching μ such that $\mu(s_i) = h_i$, for all $i \in \{1, \ldots, n\}$. Since every student has an unique highest ranked school, there is exactly one Pareto efficient matching μ^*, with $\mu^*(s_i) = h_m$, for all $i \in \{1, \ldots, n\}$. Suppose $\varphi(P) = \mu$. Define $\succeq'$ as an alteration of $\succeq$ by s_n, such that $h_1 \succ'_{s_n} h_2 \succ'_{s_n} h_n \succ'_{s_n} h_j$, for all $j \in \{3, \ldots, n-1\}$. Now, if s_n is not assigned to school h_1 or h_2 and we still maintain the cycle of size n, then there is at least one additional edge leaving from s_n. This means that there is a cycle with less than n students, which contradicts with k-coalition stability. If student s_n is assigned to h_2, then there is a cycle of less than n students as well, via the previous declaration. So, s_n must be assigned to school h_1. Since $h_1 \succ_{s_n} \mu(s_n)$, we have that $\succeq'$ is a manipulation, which contradicts with the strategy-proofness of φ. Therefore, $\varphi(P) \neq \mu$, which means that the only valid matching to produce is μ^*, which is Pareto efficient. □

A mechanism that is k-coalition stable, but not necessarily Pareto efficient is the following: Start with the random assignment μ. If there is a cycle in $G(\mu)$ that has length k or less, remove this cycle by assigning every student in the cycle to its successor in the cycle. Repeat until no cycles with length k or less remain. This mechanism looks similar to the top trading cycles mechanism, as used by Abdulkadiroglu and Sonmez [1], which is equivalent to the random SD algorithm.

However, our k-coalition stable mechanism is not strategy-proof. For the problem considered in Lemma 1, if we start with the only k-coalition stable, but not Pareto efficient matching, the mechanism will return that matching, which is manipulable, as Lemma 1 has shown.

We conclude that it is not easy to find a practical, strategy-proof, mechanism that lies in the 'gap' between k-coalition stability and Pareto efficiency, if any exist.

4 Anonymity and neutrality

Although anonymity and neutrality are interesting properties, they are hard to prove and most results are negative. Hence, it is both useful and sufficient to restrict ourselves to the cases where $c_h = q_h = 1$ for all $h \in H$, so we will make this restriction. Additionally, the mechanisms introduced in this section are of little direct practical use and we provide no efficient method to apply them. Instead, the main purpose of these mechanisms is, by exploring the theoretical boundaries of these solutions, to get a better view of the nature of the school assignment problem.

The main reason to study *anonymity* is that expresses 'equal treatment' in a precise way.

Definition 8 (Anonymity and Neutrality)**.** Let problem $P = (S, H, \succeq)$. A *relabelling* Π of students and schools is a function $\Pi : S \cup H \to S \cup H$, such that $\{\Pi(s) \mid s \in S\} = S$ and $\{\Pi(h) \mid h \in H\} = H$. Note that since H and S are finite, Π is bijective. A relabelling Π is called *trivial* if $\Pi(x) = x$ for all $x \in S \cup H$.

Define $\Pi(P)$ as equal to $(S, H, \succeq')$, such that for all students $s \in S$ and for all schools $h, i \in H$, we have $\Pi(h) \succeq'_{\Pi(s)} \Pi(i)$ if and only if $h \succeq_s i$.

Let μ be a matching. Define $\Pi(\mu)$ as equal to μ', such that for all students $s \in S$ and for all schools $h \in H$, we have $\mu'(\Pi(s)) = \Pi(h)$ if and only if $\mu(s) = h$.

A mechanism φ is *anonymous* for an assignment problem P, if for every relabelling Π that only changes the labels of the students, we have that $\varphi(\Pi(P)) = \Pi(\varphi(P))$.

A mechanism φ is *neutral* for an assignment problem P, if for every relabelling Π that changes the labels of the schools, we have that $\varphi(\Pi(P)) = \Pi(\varphi(P))$.

It is not possible to construct an anonymous mechanism for all assignment problems:

Example 2. Consider the assignment problem P, equal to $(S, H, \succeq)$, with $S = \{a, b\}$ and $H = \{1, 2\}$. The preference profile $\succeq$ is defined such that $1 \succ_a 2$ and $1 \succ_b 2$.

Let φ be a mechanism. μ is the matching such that $\mu = \varphi(P)$. The permutation Π is a relabelling of students, with $\Pi(a) = b$ and $\Pi(b) = a$. Now, we have that $\Pi(P) = P$, since the preferences of a and b are the same. However, we have that $\Pi(\mu) \neq \mu$, as Π is not trivial, so $\varphi(\Pi(P)) \neq \Pi(\varphi(P))$. This means that there exists no mechanism that is anonymous for this assignment problem.

In the previous example, we have seen that, since students a and b are indistinguishable after removing their labels, we cannot have anonymity. Note that the only property we required to show this is that there were some students with the same preference. Therefore, we have that no anonymous mechanism exists for any of these problems.

Definition 9 (Unique preferences)**.** Let P be an assignment problem equal to $(S, H, \succeq)$. P has *unique preferences* if $\succeq_s = \succeq_t \Rightarrow s = t$ for all students $s, t \in S$.

Call a mechanism *anonymous* if it is anonymous for all assignment problems with unique preferences. Anonymity is a strong property for a mechanism to have. However, there do exist anonymous mechanisms. We give a general class of these mechanisms.

Definition 10 (Class of preference based mechanisms Φ_A)**.** Let P be a problem with unique preferences and μ a feasible matching for P. Define the *signature* $\sigma(P)$ of a problem P and associated matching μ as follows: $\sigma(P, \mu) = \{(\succeq_s, \mu(s)) \mid s \in S\}$. The signature set Σ is equal to $\{\sigma(P, \mu) \mid \mu \in \mathcal{M}, P \text{ has unique preferences}\}$.

Call a mechanism φ *preference based*, if there exists a total, strict ordering over Σ such that for all feasible problems P with unique preferences, $\varphi(P) = \text{argmin}_{\mu \in \mathcal{M}} \sigma(P, \mu)$ with respect to the given ordering over Σ. Φ_A is the set of all preference based mechanisms.

For all problems P with unique preferences, the signature $\sigma(P, \mu)$ is unique for all feasible matchings μ. This means that, for a total, strict ordering on Σ, there exists a unique minimum. Therefore, there exist mechanisms that are preference based. Now, we show that mechanisms in Φ_A are indeed anonymous:

Lemma 2. *Let φ be a mechanism. If $\varphi \in \Phi_A$, then φ is anonymous.*

Proof. Let $\varphi \in \Phi_A$. If P has unique preferences, then, for a relabelling of students Π, we have that $\sigma(P, \mu) = \sigma(\Pi(P), \Pi(\mu))$, as all preferences are still associated with the same matched school, since Π is a relabelling of students. Therefore,

$$\operatorname*{argmin}_{\mu \in \mathcal{M}} \sigma(P, \mu) = \operatorname*{argmin}_{\mu \in \mathcal{M}} \sigma(\Pi(P), \Pi(\mu)) \tag{3}$$

since both P and $\Pi(P)$ have unique preferences. This means both minima have an unique associated argument, from which (3) follows. So, we have:

$$\begin{aligned} \varphi(\Pi(P)) &= \operatorname*{argmin}_{\mu \in \mathcal{M}} \sigma(\Pi(P), \mu) \\ &= \operatorname*{argmin}_{\Pi(\mu) \in \mathcal{M}} \sigma(\Pi(P), \Pi(\mu)) \\ &= \operatorname*{argmin}_{\Pi(\mu) \in \mathcal{M}} \sigma(P, \mu) \\ &= \Pi(\operatorname*{argmin}_{\mu \in \mathcal{M}} \sigma(P, \mu)) \\ &= \Pi(\varphi(P)), \end{aligned}$$

which means φ is anonymous. □

Before we show that strategy proofness is incompatible with anonymity, we require some definitions and a rather technical lemma:

Definition 11 (Isomorphic problems)**.** Call two assignment problems P, Q *isomorphic* if there exists a relabelling Π, such that $\Pi(P) = Q$. Since a relabelling is bijective and a composition of relabellings is a relabelling, being isomorphic is an equivalence relation. Define an *isomorphism class* as a set $C \subset \mathcal{P}$, such that for all problems $P, Q \in C : P$ isomorphic to Q, i.e. an equivalence class of the isomorphic relation. Let $\mathcal{C}$ be the set of all isomorphism classes. Call an assignment problem P *non-trivially automorphic*, if there exists a non-trivial relabelling Π such that $P = \Pi(P)$.

Note that if one $P \in C$ is non-trivially automorphic with relabelling Π, then for any problem $P \in C$, we have that, for some relabelling Π': $P' = \Pi'(P) = \Pi'\Pi(P)$. If the relabelling $\Pi'\Pi = I$, then $P = P'$, otherwise, $\Pi'\Pi$ is a non-trivial relabelling. So for an isomorphism class C, either all problems $P \in C$ are non-trivially automorphic, or none of them are.

Definition 12 (Alteration Stability)**.** Let $(S, H, \succeq)$ be a feasible assignment problem, such that there exists a school $h^* \in H$ such that $h^* \succeq_s h$ for all students $s \in S$ and schools $h \in H$. Let φ be a mechanism. The matching μ is equal to $\varphi(S, H, \succeq)$. Let $\succeq'$ be an alteration of $\succeq$ by some student $s' \neq \mu(h^*)$. Call this alteration *stable* for the mechanism φ, if $\mu(h^*) = \mu'(h^*)$, where $\mu' := \varphi(S, H, \succeq')$. A mechanism φ is *unstable* if there exists no stable alteration for φ.

We have been unable to determine whether there exists an unstable mechanism. However, we show that no mechanism can be both unstable and strategy-proof in Lemma 3, which is is used to prove Theorem 3.

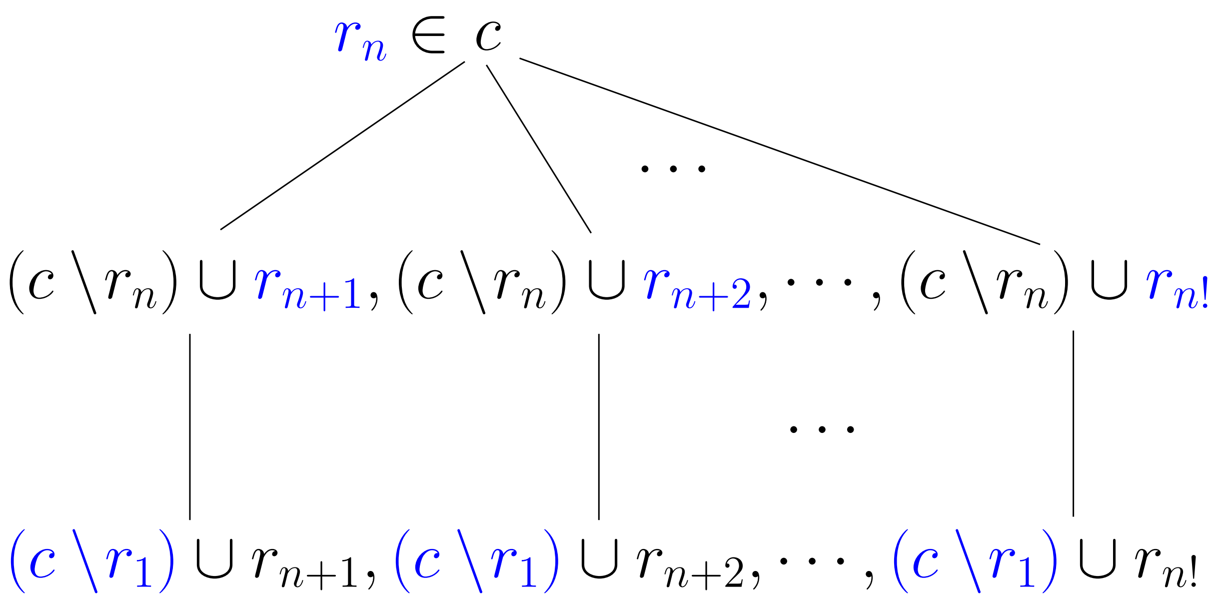

Figure 2: A diagram of a part of $\mathcal{G}(n)$, used in the proof of Lemma 3. Each vertex is denoted by a set v. The subset of v that must contain the student matched to h^* is coloured blue.

Lemma 3. *If a mechanism φ is both anonymous and strategy proof, then there exists an assignment problem with n students and with a stable alteration for φ, for all $n \geq 5$.*

Proof. Suppose φ is both anonymous and strategy proof. For the following, consider only problems with exactly n students, n schools of capacity 1, so $S = \{s_1, \dots, s_n\}$ and $H = \{h_1, \dots, h_n\}$. Additionally, every student has unique preferences and there is a school $h^* \in H$ such that $h^* \succeq_s h$ for all $s \in S$ and $h \in H$. $\mathcal{P}'(n)$ is the set of all such problems with n students. For an isomorphism class C, we define $C'(n)$ as the isomorphism class, restricted to $\mathcal{P}'(n)$, so $C'(n) = C \cap \mathcal{P}'(n)$. $\mathcal{R}(n)$ is the set of preferences a student can have in a problem in $\mathcal{P}'(n)$. Since a student can pick any permutation of the schools as its preference, as long as h^* is most preferred, we have that $|\mathcal{R}(n)| = (n-1)!$. $\mathcal{C}'(n)$ is the set of all isomorphism classes restricted to $\mathcal{P}'(n)$, equal to $\{C \in \mathcal{P}'(n) \mid C \text{ is an isomorphic class}\}$. Note that an isomorphism class can be uniquely represented by a subset of $\mathcal{R}(n)$, that is the set of the different preferences used by the students. We denote such a representation by c', for an equivalence class C'. Define the (undirected) *conflict graph* $\mathcal{G}(n)$ for a problem set $\mathcal{P}'(n)$ as follows: let $\mathcal{C}'(n)$ be the vertex set and draw an edge between $C', D' \in \mathcal{C}'(n)$ if and only if $|c' \cap d'| = n - 1$, i.e. the representations of the equivalence classes differ in exactly one preference, which means that there are problems in these classes that are alterations of each other.

Since φ is anonymous, it must assign h^* to the student with the same preference for all problems in the same isomorphism class in $\mathcal{C}'(n)$. So, for simplicity, we consider φ to assign h^* to a certain preference, for every isomorphism class in $\mathcal{C}'(n)$. If some classes C' and D' are adjacent in $\mathcal{G}$ and φ assigns h^* to the same preference in both cases, then we have a stable alteration. We will show that, for certain cases, this graph must have some adjacent equivalence classes that assign h^* to the same preference. See Figure 2 for a sketch of this process.

For ease of notation, we define r_i, for $i \in \mathbb{N}$ such that $\{r_1, \dots, r_{(n-1)!}\} = \mathcal{R}(n)$. Now, consider the equivalence class C, with representation $c := \{r_1, \dots, r_n\}$. Without loss of generality, suppose that φ assigns h^* to preference r_n. Note that c is adjacent to $(c \setminus \{r_n\}) \cup \{r_k\}$, for all $k \in \{n+1, \dots, (n-1)!\}$. For all these equivalence classes, φ assigns h^* to r_k. The reason for this is that otherwise, the student with preference r_k can alter its preference to r_n and will be assigned to h^*, which is a better school than the one originally assigned to, so this is a manipulation. This contradicts with strategy-proofness. For every k, we have that $(c \setminus \{r_n\}) \cup \{r_k\}$ is adjacent to $(c \setminus \{r_1\}) \cup \{r_k\}$. This means that φ cannot assign h^* to r_k for those equivalence classes. Note that the set of these vertices, for all k, forms a clique in $\mathcal{G}$ (i.e. they are all adjacent to each another), of size $(n-1)! - n$. Additionally, we have $n-1$ different preferences over all these vertices, for which we must try to have every equivalence class to assign h^* to a different one. Hence, we need to properly colour a clique of size $(n-1)! - n$ with $n-1$ colours. This is possible if and only if $(n-1)! - 1 \leq n - 1$, since we need to colour every vertex differently. This, however, is false for all $n \geq 5$. So, φ, given that it is anonymous and unstable, cannot be strategy proof. □

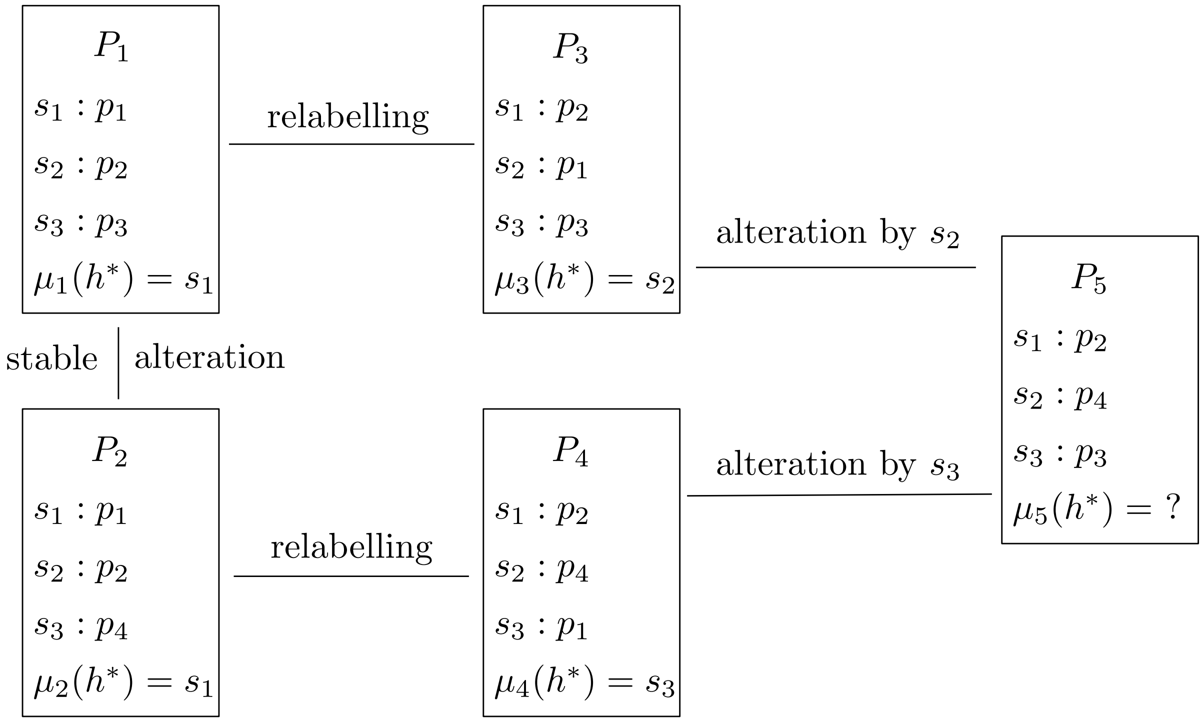

Figure 3: A diagram of the relevant differences between the assignment problems used in the proof of Theorem 3 and their relations. $s : p$ denotes that student s has preference p.

Using this lemma, we can prove a more general result:

Theorem 3. *If a mechanism is anonymous, it is not strategy proof.*

Proof. See 3 for a diagram of all relevant variables for this proof. Suppose some mechanism φ is both anonymous and strategy proof. By Lemma 3, there exists an assignment problem $P_1 = (S, H, \succeq^1)$, with a stable alteration. Let student s_1 be assigned to h^* by φ in this problem. Let $\succeq^2$ be such a stable alteration. In particular, let p_1, p_2, p_3, p_4 be distinct preferences and s_1, s_2, s_3 be distinct students, such that $\succeq^1_{s_1} = \succeq^2_{s_1} = p_1$, $\succeq^1_{s_2} = \succeq^2_{s_2} = p_2$, $\succeq^1_{s_3} = p_3$ and $\succeq^2_{s_3} = p_4$. Write $\mu_i := \varphi(P_i)$, with $P_i = (S, H, \succeq^i)$, for all $i \in \{1, \dots 5\}$. Since $\succeq^2$ is a stable alteration, we have that $\mu_1(s_1) = \mu_2(s_1) = h^*$. Let $\succeq^3$ be such that $\succeq^3_{s_1} = p_2, \succeq^3_{s_2} = p_1$ and $\succeq^3_s = \succeq^1_s$, for all $s \in S - \{s_1, s_2\}$. Let $\succeq^4$ be such that $\succeq^4_{s_1} = p_2, \succeq^4_{s_2} = p_4, \succeq^4_{s_3} = p_1$ and $\succeq^4_s = \succeq^2_s$, for all $s \in S - \{s_1, s_2, s_3\}$.

Since φ is anonymous, $\succeq^3$ is a relabelling of $\succeq^1$ and $\succeq^3_{s_2} = \succeq^1_{s_1}$, we have that $\mu_3(s_2) = \mu_1(s_1) = h^*$. Analogously, as $\succeq^4$ is a relabelling of $\succeq^2$ and $\succeq^4_{s_3} = \succeq^2_{s_1}$, $\mu_4(s_3) = \mu_2(s_1) = h^*$.

$\succeq^5$ is an alteration of $\succeq^3$ by s_2, such that $\succeq^5_{s_2} = p_4$. If $\mu_5(s_2) \neq h^*$, then $\mu_3(s_2) \succ^5_{s_2} \mu_5(s_2)$, i.e. $\succeq^3$ is a manipulation by s_2 and φ is not strategy proof. So, $\mu_5(s_2) = h^*$. However, $\succeq^4$ is also an alteration of $\succeq^5$, but by s_3. Since $\mu_5(s_2) = h^*$, $\mu_5(s_3) \neq h^*$. Then, we have that $\mu_4(s_3) = h^* \succ^5_{s_3} \mu_5(s_3)$, so now $\succeq^4$ is a manipulation by s_3. This means that φ is not strategy proof.

□

Call a mechanism *neutral* if it is neutral for all assignment problems with unique preferences. Although all mechanisms $\varphi \in \Phi_A$ are anonymous, they do not necessarily satisfy neutrality. These two properties seem intuitively conflicting: if we can neither use the names of the students for arbitration nor that of the schools, how can we solve those instances? While there is no mechanism that generally satisfies both these properties, the following class of mechanisms satisfies these properties for all assignment problem in which they are both satisfiable, as characterised by Theorem 4.

Definition 13 (Class of anonymous and neutral mechanisms Φ_{AN})**.** For all isomorphism classes C containing no non-trivially automorphic problems, let a fixed $P \in C$ be the representation of C, denoted by $p(C)$ and a fixed feasible matching μ for P be the associated matching, denoted by $\mu(C)$.

Φ_{AN} is a set of mechanisms, such that $\varphi \in \Phi_{AN}$, if and only if for any $P \in \mathcal{P}$, we have $\varphi(P) = \Pi(\mu(C))$, where C is an isomorphism class, such that $P \in C$, and Π is relabelling such that $P = \Pi(p(C))$.

Note that this is well defined, since there exists only one such C (as it is an equivalence class), there is such a relabelling Π (P and $p(C)$ are both in C) and only one such relabelling (see Definition 11). In other words, there is at least one mechanism for which this definition holds, so $\Phi_{AN} \neq \emptyset$.

We give the following characterisation for Φ_{AN}:

Theorem 4. *Let Φ be the class of mechanisms that are both anonymous and neutral for all assignment problems that are only trivially automorphic. Then, $\Phi = \Phi_{AN}$.*

Proof. '$\Phi_{AN} \subseteq \Phi$': Let mechanism $\varphi \in \Phi_{AN}$. Since P is only trivially automorphic, the following holds for any relabelling Π, Π'. If $\Pi'(P) = \Pi(P)$, then $\Pi^{-1}\Pi'(P) = P$. As $\Pi^{-1}\Pi'$ is an automorphism, it must be trivial, so $\Pi^{-1}\Pi' = I$, for the identity mapping I. Additionally, all problems P' in the isomorphism class C such that $P \in C$ are only trivially automorphic. So, for all relabellings Π: $\varphi(\Pi(p(C))) = \Pi(\mu)$. This holds, since for every problem $P \in C$, there is exactly one relabelling Π such that $P = \Pi(p(C))$. Let Π' be such that $P = \Pi'(p(C))$. Then, for all relabellings Π, we have:

$$\varphi(\Pi(P)) = \varphi(\Pi\,\Pi'(p(C))) = \Pi\,\Pi'(\mu) = \Pi(\varphi(P)).$$

So, $\varphi \in \Phi$.

'$\Phi \subseteq \Phi_{AN}$': Let $\varphi \in \Phi$. Suppose that there is a isomorphism class C with only trivially automorphic problems for which there is no representation $p(C)$ and associated matching $\mu(C) = \varphi(p(C))$, such that for all relabellings Π: $\varphi(\Pi(p(C))) = \Pi(\mu(C))$. So, there is a permutation Π, such that $\varphi(\Pi(p(C))) \neq \Pi(\mu(C)) = \Pi(\varphi(p(C)))$, so φ is not both anonymous and neutral, a contradiction. So, for all isomorphism classes with only trivial automorphic problems, we have that there exists a $p(C)$ and the associated matching $\mu(C)$, such that $\varphi(p(C)) = \mu(C)$ and that for all relabellings Π: $\varphi(\Pi(p(C))) = \Pi(\mu(C))$. So, $\varphi \in \Phi_{AN}$.

So, from the previous paragraphs, it follows that $\Phi = \Phi_{AN}$. □

5 Mechanisms for minimum quorums

There have been multiple approaches to adapt SD to include quorums.

According to step 5 in an explanation of the second algorithm used in Amsterdam [6], the algorithm checks whether the minimum number of assigned students for all classes in the schools is met. Then, a part of the algorithm is changed slightly and then re-run, until the minima are satisfied.

Another is the main result of [13], the Serial Dictatorship with Project Closures algorithm (SDPC). We introduce yet another algorithm, an adaption on SDPC, Serial Dictatorship with Non-selfish Project Closures (BSD). Before we discuss these algorithms, we present the following definition, given in [13].

Definition 14 (Active school)**.** A school $h \in H$ is considered *active* at step t of an SD based algorithm if the following conditions hold:

1. For all steps before t, the number of students assigned to h is less than or equal to its capacity c_h; and

2. In case the t-th student in the given enumeration is assigned to school h, then the remaining $n - t$ students are enough to fill the quorum of h and of all schools to which some student has been assigned in previous steps.

Furthermore, let $A_t := \{h \in H | \ h$ is active at step $t\}$, i.e. the set of active schools at step t.

Note that condition 2 of this definition is equivalent to requiring that $\sum_{h \in H'} q_h(t) \leq n - t$, where $H' = \{\mu(E(i)) \ \mid 1 \leq i \leq t\}$. Using this definition, the SDPC algorithm [13] is specified as follows:

Algorithm 3 (Serial Dictatorship with Project Closures)**.** Given an enumeration E of S, the *Serial Dictatorship with Project Closures* algorithm does the following for every step t, where $1 \leq t \leq n$, for the students $s \in S$, in the order of E: Assign s to the most preferred school of s in A_t.

Although this algorithm has been proven to be both strategy proof and Pareto efficient, there is an important issue. In the model of [13], a matching is feasible even if some students are not assigned to any school, contrary to our definition. This is, in fact, a possible outcome of SDPC, which we show in Example 3.

To find a good solution for matchings with minimum quorums in in our model, we require another definition.

Definition 15 (Benevolent school)**.** Call a set of schools $H' \subseteq H$ *attainable* at step t of a SD based mechanism, if there exists a feasible assignment of $n - t$ students to the schools in H, such that, in addition to all students being assigned to schools, all schools in H' have at least one student assigned to them. Call an assignment of a student to school h at step t of an SD based algorithm *benevolent*, if $c_h(t) > 0$ and there is an attainable set of schools H' at step t, such that for all $i \in \{1, \ldots, t\}$, we have $\mu(E(t)) \in H'$, i.e. all students assigned in earlier steps and the current step are assigned to some school in the attainable set H'.

Let $A^*_t := \{h \in H | \ h$ is benevolent at step $t\}$, i.e. the set of benevolent schools at step t.

Note that a school h is attainable if and only if

$$\sum_{h \in H'} q_h(t) \leq n - t \leq \sum_{h \in H'} c_h(t). \tag{4}$$

To see this, observe that, if Equation 4 holds, we can first assign enough students to every school $h \in H'$ such that h reaches its minimum quorum, since $\sum_{h \in H'} q_h(t) \leq n - t$. Then, if any students remain, there is enough remaining capacity to assign them to one of these schools, resulting in a feasible assignment. Additionally, if Equation 4 does not hold, then one school does not meet its quorum or has overcapacity.

Note that for a benevolent school, the first condition in Definition 14 is satisfied by definition. The second condition follows from the left inequality of Equation 4. So, we have that $A^*_t \subseteq A_t$.

Remark. It is not trivial to determine whether a set of schools H contains an attainable subset. In particular, we want to determine whether there is a subset of H for which Equation 4 holds. Fortunately, we can transform this into a well-known problem. Given is a set of schools $H = \{h_1, \ldots, h_m\}$, the total number of students n and the step t. Consider the subset $H' \subseteq H$. We will encode this in so-called decision variables $x_1, \ldots, x_m$, with $x_i \in \{0, 1\}$ and with $h_i \in H'$ if and only if $x_i = 1$. We get the following maximisation problem for these variables:

$$\text{Maximize } z = \sum_{i=1}^{m} x_i c_{h_i}(t), \quad \text{subject to } \sum_{i=1}^{m} x_i q_{h_i}(t) \leq n - t. \tag{5}$$

Now, we have that H contains a attainable subset if and only if for the maximum z, we have $z \geq n - t$. The problem stated in equation 5 is know as the *1/0-knapsack problem*, with $c(t)$ as the prices and $q(t)$ as this weights. Although knapsack is an NP-hard problem in general, there is a simple dynamic programming approach [11] that achieves a complexity of $\mathcal{O}(mW)$, where W is the maximum value of the weights. Since the maximum weight is bounded by the amount of students, solving this particular knapsack problem has a complexity of $\mathcal{O}(mn)$.

Not choosing a benevolent school means that eventually, some students end up unassigned even if another choice would allow them to be assigned. In fact, it turns out that only choosing benevolent schools is sufficient to ensure that all students are assigned in every feasible assignment problem. A natural extension to the previous algorithm would be to simply enforce such a choice. However, since a benevolent school is active, all we require is to restrict the assignments to benevolent schools.

Algorithm 4 (Benevolent Serial Dictatorship (BSD))**.** Given an enumeration E of S, the *Benevolent Serial Dictatorship* algorithm does the following for every step t, for all $1 \leq t \leq n$, for student $s = E(t)$: Assign s to the most preferred school of s in A^*_t.

Since BSD tests whether assigning a student to a school results in a benevolent matching at most $|H| \cdot |S|$ times and all those tests take $\mathcal{O}(|H| \cdot |S|)$ time, we have that BSD has a complexity of $\mathcal{O}(|H|^2|S|^2)$.

Before proving that BSD gives a feasible matching for all feasible problems, we will first show an assignment problem for which SDPC fails to do so.

Example 3. Consider the problem $P = (S, H, c, q, \succeq)$, with $S = \{s_1, s_2, s_3\}, H = \{h_1, h_2\}, q_{h_1} = 1, c_{h_1} = 2, q_{h_2} = c_{h_2} = 2$ and $h_1 \succeq_s h_2$, for all $s \in S$. We apply SPDC to P. Since all students have the same preference, we choose the enumeration E to be (s_1, s_2, s_3), without loss of generality. During the first 2 steps, all schools are active, so most preferred school of the students is chosen, which means that $\mu(h_1) = \{s_1, s_2\}$ after step 2. Now, there is no active school remaining for student s_3, which means that this student remains unassigned.

The DA-STB algorithm, which ignores minimum quorums, yields the same results. The algorithm described in [6] attempts to resolve this issue by deferring student s_3 from h_2. However, since there are still 2 students assigned to h_1, the quorum of h_2 cannot be reached, so the lack of available capacity remains. This means that this attempt is unsuccessful and the result is the same as with SDPC.

The BSD algorithm does the following. In step 1, all schools are benevolent, so s_1 is assigned to h_1. In step 2, assigning to h_1 is not benevolent, as this means that there is no attainable set of schools after this step if we assign to that school, as in the previous algorithms. So, both s_2 and s_3 get assigned to h_2, which means that no students remains unassigned.

The BSD algorithm solves this issue in general.

Theorem 5. *For a given enumeration E of the students, the BSD algorithm yields a feasible, Pareto efficient and strategy proof mechanism.*

Proof. Define mechanism BSD as the BSD algorithm for the enumeration E. Without loss of generality, we have that $P = (S, H, \succeq, c, q)$, with $n = |S|$. The matching μ is such that $\mu = \text{BSD}(P)$.

First, we confirm that BSD is well defined, i.e. that, for all feasible assignment problems, BSD makes a benevolent assignment of a student at every step. Since the problem is feasible, there is an attainable set of schools H' for all n students before the first step, so $A_1^* \neq \emptyset$. If $A_t^* \neq \emptyset$ for some $t \geq 1$, then there is an attainable set of schools $H' \subseteq H$, such that for all $i \in \{1, \dots, t\}$, we have $\mu(E(i)) \in H'$. We have that $\sum_{h \in H'} c_h(t+1) = (\sum_{h \in H'} c_h(t)) - 1 \geq n - (t+1)$, since H' is attainable at step t. If $\sum_{h \in H'} q_h(t) > 0$, then there is a school $h \in H'$ that has not met its quorum, so assigning h in step $t+1$ means that $\sum_{h \in H'} q_h(t+1) = (\sum_{h \in H'} q_h(t)) - 1 \leq n - (t+1)$. Otherwise, all schools in H' have met their quorum, so $\sum_{h \in H'} q_h(t+1) = 0 \leq n - (t+1)$. Hence, we have $\sum_{h \in H'} q_h(t) \leq n - (t+1) \leq \sum_{h \in H'} c_h(t+1)$ in either case, which means that H' is attainable at step $t+1$. Additionally, for all $i \in \{1, \dots, t+1\}$, we have that $\mu(E(i)) \in H'$, so $A_{t+1}^* \neq \emptyset$. By induction on the step t, we have that for all steps t: $A_t^* \neq \emptyset$. Therefore, BSD is well defined.

Feasibility: The sound definition of BSD, yields that $|\mu(s)| = 1$, for all students $s \in S$. Since all assignments are benevolent, no student is assigned to a school h with $c_h(t) = 0$, so $|\mu(h)| \leq c_h$, for all schools $h \in H$.

Consider step n. The set of schools H^* with some student assigned to them is equal to $\{h \in H | \mu(h) \neq \emptyset\}$. Since the assignment in this step is benevolent, we have that there exists an attainable set of schools H' for step n, such that $\mu(E(i)) \in H'$ for all $i \in \{1, \dots, n\}$, so $H^* = \{E(i) | 1 \leq i \leq n\} \subseteq H'$. Since all schools in H' have at least one student assigned to them, we have that $H' \subseteq H^*$, so $H^* = H'$. Since H' is attainable, we have that $\sum_{h \in H^*} q_h(n) = \sum_{h \in H'} q_h(n) \leq n - n = 0$, so the minimum quorum of all schools to which students are assigned is reached, i.e. $|\mu(h)| \geq q_h$ for all schools $h \in H^*$. Since all other schools have no students assigned to them, μ is feasible. Since P is an arbitrary feasible assignment problem, we have that BSD is a feasible mechanism.

Pareto efficiency: The proof of this property is similar to that of Theorem 1. Suppose there exists some feasible matching $\bar{\mu}$ that Pareto dominates μ. This means there exist $s \in S$ such that $\bar{\mu}(s) \succ_s \mu(s)$. Let s^* be the first such student, in the order given by E, and let $h^* := \bar{\mu}(s^*)$. So, since s^* prefers h^* over $\mu(s^*)$, there either was no remaining capacity at school h^* when the BSD algorithm reached s^* or h^* is not benevolent at step $E(s^*)$. If h^* is not benevolent at $E(s^*)$, we have that there is some student s for which $|\bar{\mu}(s)| = 0$. This means $\bar{\mu}$ is not feasible, which is a contradiction. Otherwise, since s^* is assigned to h^* in $\bar{\mu}$, there is a student $s' \neq s^*$ with $\mu(s') = h^*$ and $\bar{\mu}(s') \neq h^*$. Since $\bar{\mu}$ Pareto dominates μ, we have that $\bar{\mu}(s') \succ_{s'} \mu(s')$. However, since $\mu(s') = h^*$, we have that s' occurs before s^* in E, which contradicts s^* being the first such student. So, there exists no matching that Pareto dominates μ and μ is feasible, which means that μ is Pareto efficient. Since the assignment problem P was arbitrary, BSD is Pareto efficient.

Strategy proofness: Again, the proof of this property is similar to that of Theorem 1. Let $P' := (S, H, \succeq')$, with $\succeq'$ an alteration of $\succeq$ by some student s. Let $\mu' := \text{BSD}(P')$. Since all students other than s have the same preferences as in P and whether a school is beneficial is not influenced by any preference, the BSD algorithm has the same result for all students assigned before s. So, the set of beneficial schools for s is the same in both cases. Since BSD assigns to s the best of these schools, according to the given preference, we have that $\mu(s) \succeq_s \mu'(s)$, since for P, BSD assigns the optimal school of this set under $\succeq$. So, $\succeq'$ cannot be a manipulation and since $\succeq'$ is arbitrary, no manipulation is possible, which means BSD is strategy proof. □

Example 4. For any $n \in \mathbb{N}$, consider the assignment problem $(S, H, \succeq, c, q)$, with $S = \{s_1, \ldots, s_n\}$, $H = \{h_1, h_2\}$, $q_1 = c_1 = 1$, $q_2 = c_2 = n$ and for all students $s \in S : h_1 \succ_s h_2$. If we apply SDPC on this problem, with any enumeration of the students, the first student is allowed to pick both h_1 and h_2, as both are active schools and no schools with students assigned to them are made inactive by either choice. So, as any student prefers h_1, the first student is assigned to school h_1, filling the capacity. h_2 is now inactive, since there are not enough remaining students to reach the minimum of n. So, all other students remain unassigned, which means that this solution is not a feasible matching. In fact, it is clear that assigning all students to school h_2 is the only feasible matching, so the BSD algorithm returns this matching.

Note that the BSD algorithm gives the same results as SDPC and the algorithm used in Amsterdam, for all problems where those mechanisms give a feasible matching.

6 Interpretation and conclusion

In this section, we will discuss the relevance and applicability of our results for the scenario in Amsterdam.

We considered the rank minimizing algorithm, which attains the maximum efficiency of a result at the cost of losing the strong property of strategy-proofness. Strategy-proofness is a strong property, since it could be the case that, in practice, a student needs more information than he/she has available to actually make a manipulation. For example, we can expect that a student has only partial information on the preferences of other students. Additionally, a student likely has no information on the outcome of randomisation steps in the algorithm.

We have shown that there is a practical polynomial-time algorithm to construct a manipulation for the rank minimizing algorithm, but the algorithm uses a significant amount of knowledge to do this effectively. So, we suspect that, under some model which takes information into account, it can be shown that constructing a manipulation is too uncertain, such that the probability of a successful manipulation is too low to outweigh the risk of being worse of after misrepresenting their own preference. There is some global data publicly available on the preferences of previous years [10]. Although the data is similar between the years, it does not contain the specific preferences of all students, which is relevant for the rank minimizing algorithm.

Since we have no clear results on the manipulability of the rank minimizing algorithm, there is a reasonable chance that a significant amount of students will not give their true preference. Therefore, we think that this algorithm is currently unsuitable for usage in Amsterdam. However, if a good argument against practical manipulability can be made, this algorithm becomes interesting, as its efficiency would be greater than the SD algorithm.

Motivated by a complete lack of 'equal treatment' of students in the SD mechanism, we studied the consequences of requiring equal treatment, formalised as anonymity. Apart from the trivially impossible cases, we have shown that, when restricting to unique preferences for every student, this requirement cannot be satisfied simultaneously with strategy proofness. So, in other words, some form of arbitration independent of the assignment problem is required to satisfy strategy proofness. This result motivates the idea that, at least theoretically, the fixed enumeration in the SD algorithm is a necessary evil and therefore determining this enumeration uniformly at random is the next best form of equal treatment. This reinforces the idea that the the (R)SD algorithm is fair.

For the case with minimum quorums, we have provided the BSD algorithm, as a generalisation of the SD algorithm. This is a generalisation, in the following sense. For a certain enumeration, if we have that the SD algorithm, applied to the minimum quorum case, results in a feasible matching, the BSD algorithm with the same enumeration yields the same matching. Additionally, the main properties of SD, strategy-proofness and Pareto efficiency, hold for the BSD algorithm in general. The SDPC by Monte and Tumennasan [13] is a generalisation of the SD in the same vein, but for their definition of feasibility, which allows for unassigned students.

This difference is also the only difference between BSD and SDPC. Note that BSD algorithm can be applied in the model of Monte and Tumennasan as well, since our model is more restrictive. For the Amsterdam problem the BSD algorithm is the better choice, since all students should be eventually assigned to some school.

The "SD with minimum quotas" as described Fragiadakis et al. [4] is an extension of the SD as well, in the same sense as the earlier two, but under a more restrictive model, as all school must have some student assigned to them. This can be considered as a special case of the BSD or SDPC, where we determine the final attainable set of schools, before the algorithm, not necessarily depending on the student preferences. For the Amsterdam problem, it is strange to determine the final attainable set of schools before running the algorithm, thereby restricting the students choices, as we do not need to assign a student to every school. So, the BSD algorithm is preferred.

This means that we have offered an algorithm, which is more suitable for the Amsterdam scenario with minimum quorums than the existing algorithms.

References

[1] Atila Abdulkadiroglu and Tayfun Sonmez, *Random serial dictatorship and the core from random endowments in house allocation problems*, Econometrica **66** (1998), no. 3, 689 (English).

[2] David J. Abraham, Katarina Cechlarova, David F. Manlove, and Kurt Mehlhorn, *Pareto optimality in house allocation problems* **3827** (2005), 1163–1175 (English).

[3] Lars Ehlers and Bettina Klaus, *Strategy-proofness makes the difference: Deferred-acceptance with responsive priorities*, Mathematics of Operations Research **39** (2014), no. 4, 949–966.

[4] Daniel Fragiadakis, Atsushi Iwasaki, Peter Troyan, Suguru Ueda, and Makoto Yokoo, *Strategyproof matching with minimum quotas*, ACM Trans. Econ. Comput. **4** (January 2016), no. 1, 6:1–6:40.

[5] David Gale and Lloyd S. Shapley, *College admissions and the stability of marriage*, The American Mathematical Monthly **69** (1962), no. 1, 9–15.

[6] Pieter A. Gautier, Monique de Haan, Bas van der Klaauw, and Hessel Oosterbeek, *Matching in het voortgezet onderwijs Amsterdam*, 2014 (dutch). URL: `www.verenigingosvo.nl/wp-content/uploads/2014/09/Uitleg-Matching.pdf` (Retrieved on March 2016).

[7] Pieter A. Gautier, Monique de Haan, Bas van der Klaauw, and Hessel Oosterbeek, *Schoolkeuze voortgezet onderwijs in Amsterdam: Verslag van een simulatiestudie*, VU Amsterdam, 2014 (dutch).

[8] Pieter A. Gautier, Monique de Haan, Bas van der Klaauw, and Hessel Oosterbeek, *Eerste analyse matching en loting voortgezet onderwijs Amsterdam 2015*, VU Amsterdam, 2015 (dutch).

[9] Pieter A. Gautier, Monique de Haan, Bas van der Klaauw, and Hessel Oosterbeek, *The performance of school assignment mechanisms in practice*, IZA Discussion Papers **9118** (2015) (English).

[10] Pieter A. Gautier, Monique de Haan, Bas van der Klaauw, and Hessel Oosterbeek, *Eerste analyse matching en loting voortgezet onderwijs Amsterdam 2016*, VU Amsterdam, 2016 (dutch).

[11] Hans Kellerer, Ulrich Pferschy, and David Pisinger, *Knapsack problems*, Springer, Berlin, 2004 (English).

[12] Harold W. Kuhn, *The Hungarian method for the assignment problem*, Naval Research Logistics (NRL) **52** (2005), no. 1, 7–21.

[13] Daniel Monte and Norovsambuu Tumennasan, *Matching with quorums*, Economics Letters **120** (2013), no. 1, 14–17.

[14] Gerhard J. Woeginger, *Exact algorithms for NP-hard problems: A survey* (2001), 185–208.