

Stable Matching under Matroid Rank Valuations

Alon Eden¹, Vignesh Viswanathan² and Yair Zick²

¹Hebrew University of Jerusalem, Israel

²University of Massachusetts Amherst, USA

alon.eden@mail.huji.ac.il, {vviswanathan,yzick}@umass.edu

Abstract

We study a two-sided matching model where one side of the market (hospitals) has combinatorial preferences over the other side (doctors). Specifically, we consider the setting where hospitals have matroid rank valuations over the doctors, and doctors have either ordinal or cardinal unit-demand valuations over the hospitals. While this setting has been extensively studied in the context of one-sided markets, it remains unexplored in the context of two-sided markets. When doctors have ordinal preferences over hospitals, we present simple sequential allocation algorithms that guarantee stability, strategyproofness for doctors, and approximate strategyproofness for hospitals. When doctors have cardinal utilities over hospitals, we present an algorithm that finds a stable allocation maximizing doctor welfare; subject to that, we show how one can maximize either the hospital utilitarian or hospital Nash welfare. Moreover, we show that it is NP-hard to compute stable allocations that approximately maximize hospital Nash welfare.

1 Introduction

Stable matching is a fundamental problem in the EconCS community. In the classical version, we have a set of n doctors and n hospitals, each with ordinal preferences over the other side. The goal is to find a stable matching of doctors to hospitals; informally, a matching is stable if no doctor-hospital pair prefers each other to their assigned match. This problem has been studied since the 1950s and is well understood today. There exists a polynomial time algorithm (the Gale-Shapley algorithm) [14] that outputs a stable matching. Moreover, the set of stable matchings have a lattice structure [23]; as a consequence, there exists a stable matching that provides all doctors their best possible outcome among all the stable matchings.

Moving beyond the classical problem, several natural generalizations have been studied in the literature. One line of generalizations has been the many-to-one matching problem where a hospital can be matched with (or allocated to) multiple doctors; this is the model used by National Residency Matching Program [31]. Gale and Shapley [14] propose a model motivated by college admissions where each hospital has a cardinality constraint upper bounding the number of doctors they can be matched with. However, if we assume that each hospital recruits several doctors, it is reasonable to assume that hospitals would want residents trained in a diverse set of specialties; therefore, typically, hospitals have preferences over *sets* of doctors. Indeed, recent work builds on this model by adding structure to the acceptable sets of doctors for each hospital. In the CS community, Fleiner [12] studies a model where each hospital has a matroid constraint dictating the set of doctors they can be allocated. In the economics community, Hatfield and Milgrom [16] present a similar constraint based on the assumption that doctors are substitutes. For all these models, Gale-Shapley-style algorithms have been shown to output stable allocations. Moreover, these problems admit a similar lattice structure over the set of stable allocations, under the assumption that the preferences of each side over the other forms a total order; that is, there are no ties in the preferences.

A second line of generalization considers ordinal preferences with ties and incompleteness. In most cases, a stable matching can be trivially computed by breaking ties arbitrarily. The goal of this line of research is instead to compute a stable matching that maximizes the matching size [27, 10, 30]. This

problem is surprisingly NP-hard even in the classical one-to-one matching setting [15, 26, 19]. More generally, when preferences have ties and can be incomplete, stable matchings are no longer guaranteed to have a lattice structure.

In this paper, we present and study a natural problem at the intersection of these two lines of research. Following the model of Fleiner [12], we assume there is a matroid constraint for each hospital that their allocated set of doctors must satisfy. However, we assume that there is no preference order that hospitals have over the doctors, and that the goal of each hospital is simply to maximize the number of doctors they are allocated subject to the matroid constraint. In more technical terms, we assume hospitals have *matroid rank valuations*. This class of preferences has recently gained popularity in the one-sided markets literature due to its expressivity and the fact that the matroid structure can be leveraged to design fair, efficient and strategyproof mechanisms [2, 3, 4, 39, 5]. *This paper is the first to extensively study matroid rank valuations in two-sided markets.*

Mathematically, the class of matroid rank valuations is simple enough that the existing impossibility results for hospital strategyproofness do not apply [34]. However, this class of preferences does not admit the lattice structure that previous models exploited to understand stable matchings. This makes it an interesting class of preferences to study with many open questions. We focus on the design of *strategyproof* mechanisms that output *stable matchings* with high *market efficiency*. We study two variants of doctor utilities, resulting in two fundamental research questions.

- (Q1) Can we design strategyproof mechanisms that output stable matchings with high market efficiency, when doctors have ordinal preferences?
- (Q2) Can we compute stable allocations that maximize various welfare objectives for either side of the market (or both simultaneously), when doctors have cardinal preferences?

Both questions have been studied in most many-to-one matching models. Roth [34] shows that it is impossible to guarantee hospital strategyproofness when hospitals have ordinal preferences over the doctors along with a cardinality constraint. Hatfield and Milgrom [16] present a doctor-strategyproof algorithm for a general class of preferences which outputs the optimal stable allocation for all the doctors¹; given the earlier impossibility result, this is the best result possible.

When the set of stable matchings has a lattice structure, the best outcome for one side is well-defined and can be computed by a variant of the Gale-Shapley algorithm (see [16] and [12] for examples). Moreover, this is the worst outcome for the other side. However, with no clear lattice structure, as in our model, it is unclear what objectives can be maximized subject to stability. Addressing these questions with respect to matroid rank valuations offers useful insight into stable allocations in the absence of a clear lattice structure, as well as provides tools to design mechanisms in these settings.

1.1 Our Contributions and Techniques

Addressing (Q1) We present two mechanisms. Our first mechanism, called the high welfare serial dictatorship (HWSD), outputs a stable allocation that maximizes hospital utilitarian welfare and is strategyproof for the doctors. Moreover, when hospitals have binary OXS valuations, a subclass of matroid rank valuations [29], the mechanism is 2-approximately strategyproof for the hospitals (Theorem 3.1). Note that in our setting, hospital utilitarian welfare is equal to matching size, a notion of efficiency studied in prior work [27, 10].

HWSD works as follows: doctors take turns picking a hospital. Each doctor picks the highest-ranked hospital that is willing to accept them, i.e., accepting a doctor will not violate the hospital’s matroid constraint; however, this match is allowed only if there exists a *completion* of the resulting partial

¹In the many-to-one matching model, when there is a lattice structure over the set of stable matchings, the optimal stable allocation for the set of doctors trivially maximizes the size of the matching.

allocation that maximizes hospital utilitarian welfare.

Stability, doctor strategyproofness and max hospital utilitarian welfare follow straightforwardly from the description of the algorithm. The main non-trivial result we show for HWSO is the hospital strategyproofness guarantee. To show approximate hospital strategyproofness, we characterize the optimal misreport that any hospital can make. We use this to show that no hospital can gain significantly by misreporting. This broad proof structure has been used to show exact strategyproofness in both fair division [2] and many-to-one matching [16]. However, to show approximate strategyproofness, we analyze allocations via *reversible path augmentations* (Lemmas B.7 and B.8). Reversible path augmentations build on the technique of path augmentations used in prior work [2] to show strategyproofness with matroid rank valuations in the one-sided setting. The key difference in our proof is that we need to take doctor preferences into account in the two-sided setting; this makes our analysis quite intricate.

We also show that the same analysis can be used to prove *exact hospital strategyproofness* of the HWSO mechanism when hospitals have *binary capped additive valuations*.

The second mechanism we study is the popular serial dictatorship (SD) mechanism. Here, doctors may pick any hospital that is willing to admit them, regardless of whether the resulting partial assignment can be completed to a utilitarian welfare optimal assignment. SD is strategyproof for doctors, 2-approximately strategyproof for hospitals for *general matroid rank valuations*, and 2-approximately maximizes hospital utilitarian welfare (Theorem 3.3). Here again, the most non-trivial result is showing approximate hospital strategyproofness. To prove this, we use the simplicity of the serial dictatorship mechanism to upper bound the change in the doctors hospitals are assigned due to misreports. The analysis involves a careful accounting of the set of hospitals who were denied a match with doctors due to some hospital misreporting, and the set of doctors whose allocation changed due to misreporting.

Addressing (Q2). We focus on both the utilitarian and the Nash welfare objectives for both hospitals and doctors. The HWSO mechanism described above efficiently finds a stable outcome that maximizes hospital (unconstrained) utilitarian welfare. We complement this result by showing, rather unfortunately, that finding a stable allocation that maximizes Nash welfare for hospitals is NP-hard for any finite approximation factor (Theorem 4.4).

Moving to cardinal doctor utilities, we present an efficient algorithm that outputs a stable allocation maximizing doctor utilitarian welfare (Lemma 4.1) or doctor Nash welfare. Moreover, we show that subject to maximizing doctor utilitarian welfare (or doctor Nash welfare), we can efficiently maximize hospital Nash welfare (Theorem 4.3). This is the most technically involved result of the paper.

Maximizing doctor utilitarian welfare can be reduced to the problem of finding a max weight independent set at the intersection of two matroids. Maximizing hospital Nash welfare subject to maximizing doctor utilitarian welfare can be done via a local search algorithm. Most algorithms for matroid intersection use path augmentations in the *matroid exchange graph* [7, 8]. The proof for our local search algorithm’s correctness also follows from an analysis of the matroid exchange graph. Specifically, we show that we can find set of cycles in the exchange graph where at least one such cycle is an augmenting path with some desirable properties. We then show that repeatedly finding these desirable cycles and augmenting along them eventually increases the hospital Nash welfare without reducing doctor utilitarian welfare. This analysis generalizes existing results in the fair division literature², and may be of independent interest.

Main open question. The main open question left by our work is “Does there exist a hospital-strategyproof mechanism that outputs a stable allocation when hospitals have matroid rank valuations?”

²Specifically, we generalize the computational result of Babaioff et al. [2] who show that a max Nash welfare allocation can be computed in the one-sided setting when agents have matroid rank valuations.

1.2 Additional Related Work

Variants of the many-to-one matching problem have been studied in the literature, both by the computer science and the economics community. In the economics literature, the college admissions problem [14] was generalized to the job matching problem [22, 21, 32], and later generalized to the stable contract matching problem [16]. Both the job matching and the contract matching problem use some natural definitions of substitutes to constrain hospital preferences. For all three problems, Gale-Shapley style algorithms compute a stable matching. The college admissions and the job matching problems have also been generalized to the many-to-many matching setting, with similar results on the existence and computability of stable allocations [33, 35]. Existing results on hospital strategyproofness (or lack thereof) and manipulability of the college admissions problem [34, 37, 36] carry over to all the models described above.

The computer science community studies similar generalizations. The key difference being that the class of preferences used are described by mathematical objects (like matroids) rather than economic principles (like substitutability). Fleiner [12] generalizes the classical stable matching problem to the *matroid kernel problem*. The matroid kernel problem is a many-to-many matching problem where each hospital (resp. doctor) has ordinal preferences and a matroid constraint over the set of doctors (resp. hospitals). A Gale-Shapley style algorithm outputs stable allocations for this problem as well. This model has been further generalized to handle lower quotas on each hospital's bundle [17, 40, 13], as well as ties in the preferences [10, 20].

The problem of matching with ties in preferences has also received considerable attention. Irving [18] introduces the problem and provides definitions for stability in the presence of ties. Manlove [25] and Roth [32] present examples where one-to-one matching instances do not have a hospital-optimal (weakly) stable matching; this provides evidence of the loss of structure in the problem when ties are introduced. Manlove et al. [26] show that generalizing beyond ties to incomplete preferences with ties renders many previously easy problems NP-hard, even in the one-to-one matching setting. A specific problem of interest is finding the maximum size stable matching, with the best known approximation ratio being 1.5 [30, 15, 27].

Our problem can be seen as a special case of the *matroid kernel with ties problem*. Two papers study the matroid kernel with ties problem. Csáji et al. [10] present a 1.5-approximation algorithm for the maximum size stable matching problem in this model, generalizing the 1.5-approximation of McDermid [27], and show a matching lower bound. Kamiyama [20] studies the existence and computation of super-stable allocations in this model. Allocations are super-stable if moving any doctor to some hospital does not weakly improve both agents' outcomes; that is, it strictly worsens the outcome of one of the agents involved. Our focus is instead on weak stability (as defined by Irving [18]). Since hospitals do not have a preference order over doctors in our model, we believe this to be a more meaningful stability notion.

In recent independent work, Aziz et al. [1] consider the many-to-one matching setting, and show that it is possible to compute a stable matching which maximizes the matching size and is strategyproof for all agents involved *when hospitals have binary capped additive valuations*. In Corollary 3.2, we present a similar result. However, we note that their result is a little more general since they show that their algorithm satisfies additional properties like non-bossiness and Pareto optimality.

2 Preliminaries

We use $[k]$ to denote the set $\{1, 2, \dots, k\}$. Given a set S and an element d , we use $S + d$ and $S - d$ to denote $S \cup \{d\}$ and $S \setminus \{d\}$ respectively.

We have a set of n hospitals $H = [n]$, and a set of m doctors $D = \{d_1, \dots, d_m\}$. Each hospital $h \in H$

has a *matroid rank valuation* function $v_h : 2^D \rightarrow \mathbb{Z}$ over the set of doctors; for any set of doctors $S \subseteq D$, $v_h(S)$ denotes the value of the set of doctors $S \subseteq D$ according to the hospital h . We use $\Delta_h(S, d) = v_h(S + d) - v_h(S)$ to denote the *marginal gain* of adding the doctor d to the set S according to hospital h . A valuation function v_h is a matroid rank function (MRF) if it satisfies the following three properties: (a) $v_h(\emptyset) = 0$, (b) for each $S \subseteq D$, $d \in D \setminus S$, we have $\Delta_h(S, d) \in \{0, 1\}$, and (c) for each $S \subseteq T \subseteq D$, $d \in D \setminus T$, $\Delta_h(S, d) \geq \Delta_h(T, d)$. MRFs have the following useful property.

Observation 2.1 (Matroid augmentation property [28]). Let v_h be a matroid rank function over a set D of elements. For every $S, T \subseteq D$ such that $v_h(S) < v_h(T)$, there exists an element $d \in T \setminus S$ such that $\Delta_h(S, d) = 1$.

We assume every doctor $d \in D$ has a strict and complete preference order $\succ_d$ over the set of hospitals H ; we write $h_1 \succ_d h_2$ if the doctor d prefers the hospital h_1 to h_2 . We define cardinal preferences for doctors in Section 4.

An *allocation* $X = (X_0; X_1, \dots, X_n)$ is an $(n + 1)$ -partition of the set of doctors D . Each hospital h is allocated X_h and X_0 denotes the set of unallocated doctors. We use $X(d)$ to denote the hospital that doctor d was allocated to in the allocation X . We use the term allocation instead of matching to differentiate our setting from the one-to-one matching setting. Given an allocation X , the *utility* of hospital h is the value $v_h(X_h)$.

When analyzing the time complexity of our mechanisms, we assume we can efficiently make *value queries* to the hospital valuation functions. That is, we assume we can compute the value $v_h(S)$ for any hospital h and set of doctors S in polynomial time.

As described in the introduction, we assume each hospital's matroid rank valuation function enforces a constraint over the set of doctors the hospital can be allocated. Specifically, we assume each hospital h can only be allocated sets of doctors which correspond to independent sets in the matroid defined by the rank function v_h . This is equivalent to the notion of *non-redundancy* used in fair division [4].

Definition 2.2 (Non-redundancy). An allocation X is said to be *non-redundant* if for all $h \in H$, $v_h(X_h) = |X_h|$. A bundle $S \subseteq D$ is said to be non-redundant with respect to v_h (for some hospital h) if $v_h(S) = |S|$.

The main desideratum we require of our allocations is *stability*. To define stability, we first define a *blocking pair*. We use the same definition of blocking pair as Hatfield and Milgrom [16] and Csáji et al. [10].

Definition 2.3 (Blocking Pair). Given an allocation X , a pair consisting of a set of doctors S and hospital h (denoted (h, S)) form a *blocking pair* if:

- (a) S is non-redundant with respect of v_h and $v_h(S) > v_h(X_h)$, and
- (b) for all $d \in S$, $h \succeq_d X(d)$ with equality holding only if $h = X(d)$.

In other words, the hospital h *strictly* prefers the set of doctors S to their assigned bundle X_h , and the doctors in S all *weakly* prefer h to their assigned hospital under X . This definition naturally extends to preference orders with ties as well. When doctors' preference orders have ties in them, the above definition requires the set of doctors in S not already allocated to h to strictly prefer h to their current allocation $X(h)$.

Definition 2.4 (Stability). An allocation X is *stable* if it is non-redundant and has no blocking pair.

When agents have matroid rank functions, this condition can be simplified. We repeatedly use the following observation when proving stability guarantees (proof in Appendix A).

Observation 2.5. Given a non-redundant allocation X , a blocking pair exists if and only if there exists some doctor d and hospital h such that $\Delta_h(X_h, d) = 1$ and $h \succ_d X(d)$.

We are interested in stable allocations with welfare guarantees. An allocation X maximizes hospital *utilitarian welfare* (hospital-USW for short) if it maximizes $\text{USW}(X) = \sum_{h \in H} v_h(X_h)$. An allocation X maximizes hospital *Nash welfare* (hospital-NSW for short) if it maximizes $\text{NSW}(X) = \prod_{h \in H} v_h(X_h)$. An allocation X is an α approximation to hospital utilitarian welfare for $\alpha \geq 1$ if $\text{USW}(X) \geq \max_{X'} \text{USW}(X')/\alpha$. We can define these welfare functions for doctors similarly, assuming doctors have cardinal utility functions. When hospitals have matroid rank valuations, every stable allocation approximately maximizes hospital-USW (proof in Appendix A).

Lemma 2.6. *When hospitals have matroid rank valuations, every stable allocation is a 2-approximation of the maximum hospital-USW.*

Strategyproofness is a key desideratum in mechanism design. We assume that agent valuations are private information, and they report them to the mechanism. We say that agent i has an *incentive to misreport* their preferences if there exists a valuation $\hat{v}$ such that if agent i reports $\hat{v}$ instead of their true valuation v_i the mechanism outputs an outcome that agent i strictly prefers. A mechanism is *strategyproof* if no agent has an incentive to misreport their preferences.

We distinguish between hospital and doctor strategyproofness. A mechanism is hospital (resp. doctor) strategyproof (in short, hospital-SP/doctor-SP) if no hospital (resp. doctor) has an incentive to misreport their preferences.

While we present several doctor-SP mechanisms that yield a stable allocation, identifying mechanisms that are hospital-SP and output stable allocations turns out to be a more demanding task. We devise mechanisms to compute stable allocations that are only approximately hospital-SP. More formally, a mechanism is α -approximate hospital-SP for $\alpha \geq 1$, if for any hospital h , misreporting preferences results in a utility of at most α times the utility received under a truthful report. This notion is sometimes called approximate incentive compatibility [38].

2.1 Preference Classes

We also present some results for subclasses of matroid rank valuations. A valuation function v_h is said to be *binary OXS* if there is an unweighted bipartite graph $G = (D \cup R, E)$ for some set R such that for every $S \subseteq D$, $v_h(S)$ is equal to the size of the maximum size matching in the subgraph of G induced by the set of vertices $S \cup R$. The set R is sometimes called the set of *slots* for valuation function v_h .

A valuation function v_h is said to be *binary capped additive* if for some positive integer b_h and all $S \subseteq D$, $v_h(S) = \min\{b_h, \sum_{d \in S} v_h(\{d\})\}$. The following hierarchy relates the three valuation classes studied in this paper:

$$\text{binary capped additive} \subset \text{binary OXS} \subset \text{matroid rank}.$$

2.2 Impossibility for Hospital-USW Maximizing Hospital-SP mechanisms

The first result we present is a simple impossibility result. We show that no hospital-SP mechanism that outputs a stable allocation can be hospital-USW maximizing. This example also highlights the fact that there may be no hospital-optimal (or doctor-optimal) stable allocation in our setting.

Example 2.7. The instance consists of two hospitals $\{h_1, h_2\}$ and three doctors $\{d_1, d_2, d_3\}$. The matroid rank valuation functions for the hospitals are given as follows:

$$\begin{aligned} v_{h_1}(S) &= |S \cap \{d_2\}| + \min\{|S \cap \{d_1, d_3\}|, 1\} \\ v_{h_2}(S) &= |S \cap \{d_3\}| + \min\{|S \cap \{d_1, d_2\}|, 1\}. \end{aligned}$$

The preferences of the doctors are given as follows: d_1 and d_3 prefer h_1 to h_2 , and d_2 prefers h_2 to h_1 . There are only two hospital-USW maximizing stable allocations in this instance denoted by X and Y :

$$\begin{aligned} X_{h_1} &= \{d_2, d_3\} & X_{h_2} &= \{d_1\} \\ Y_{h_1} &= \{d_1\} & Y_{h_2} &= \{d_2, d_3\} \end{aligned}$$

Note that if h_2 were to report the valuation $v'_{h_2}(S) = |S \cap \{d_2, d_3\}|$, then the only hospital-USW maximizing stable allocation is Y . This is because (a) any stable allocation must allocate d_2 to h_2 , and (b) h_1 accepts d_1 or d_3 but not both. Using a similar argument, we can show that if h_1 were to report the valuation function $v'_{h_1}(S) = |S \cap \{d_2, d_3\}|$, then the only hospital-USW maximizing stable allocation is X .

Let $\mathcal{A}$ be a mechanism that always outputs a hospital-USW maximizing stable allocation. On the instance described above, it must either output X or Y . If it outputs X , then h_2 can deviate and become strictly better off. If it outputs Y , then h_1 can deviate and become strictly better off. Therefore, $\mathcal{A}$ cannot be strategyproof.

While this example does not completely rule out the possibility of hospital-SP mechanisms, it highlights the difficulty in the problem. The only other stable allocation for the instance above which is not X or Y is Z where $Z_{h_1} = \{d_3\}$ and $Z_{h_2} = \{d_2\}$. In order to guarantee hospital strategyproofness, a mechanism must deliberately output the low hospital-USW allocation Z either when h_1 reports truthfully or when h_1 misreports.

3 Doctors with Ordinal Preferences

Our objective is to identify doctor/hospital strategyproof mechanisms that output stable allocations, with additional hospital-USW guarantees. However, the counterexample in Section 2.2 shows that the ideal version of this result cannot be achieved. Instead, we seek mechanisms that achieve these guarantees *approximately*. We present two such mechanisms, each providing a different trade-off between hospital-USW and hospital strategyproofness.

3.1 High Welfare Serial Dictatorship (HWSD)

We present a mechanism that outputs a stable allocation, is doctor-SP and obtains optimal hospital-USW. The HWSD mechanism operates in a sequential fashion, where at each iteration, we finalize the allocation of one doctor. More specifically, at each iteration, we pick a doctor d and allocate it to its highest ranked hospital h that satisfies the following two conditions: (a) the utility of hospital h increases after it receives doctor d , and (b) the current allocation can be augmented to a hospital-USW maximizing allocation. The mechanism is presented in Algorithm 1. Missing proofs of this section appear in Appendix B.

Theorem 3.1. *The high welfare Serial Dictatorship (HWSD) mechanism has the following properties:*

- (i) *The mechanism is doctor-SP.*
- (ii) *The algorithm runs in polynomial time.*
- (iii) *The output allocation maximizes hospital-USW.*
- (iv) *The output allocation is stable.*
- (v) *The mechanism is 2-approximately hospital-SP when hospitals have binary OXS valuations.*

Note that our strategyproofness analysis is tight since Example 2.7 describes an instance where all hospitals have binary OXS valuations; therefore, any mechanism that outputs hospital-USW maximizing

ALGORITHM 1: High Welfare Serial Dictatorship (HWSD) Mechanism

```
 $X = (X_0, X_1, \dots, X_n) \leftarrow (D, \emptyset, \dots, \emptyset)$ 
for  $i$  in 1 to  $m$  do
  for  $j$  in 1 to  $n$  do
    Let  $h$  be the  $j$ -th highest ranked hospital for doctor  $d_i$ 
    if  $\Delta_h(X_h, d_i) = 1$  then
      if there exists a Max hospital-USW allocation  $Y$  such that  $X_{h'} \subseteq Y_{h'}$  for all  $h' \neq h$  and
         $X_h + d \subseteq Y_h$  then
           $X_h \leftarrow X_h + d$ 
           $X_0 \leftarrow X_0 \setminus d$ 
          Break
    return  $X$ 
```

allocations can only be at most 2-approximately hospital-SP. Restricting hospital preferences further to binary capped additive valuations, we show that the HWSD mechanism is exact strategyproof for hospitals.

Lemma 3.2. *When hospitals have binary capped additive valuations, the HWSD mechanism is hospital-SP.*

3.2 Serial Dictatorship

Let us next explore the Serial Dictatorship mechanism (Algorithm 2). This is another simple mechanism which offers (weaker) approximate welfare and (stronger) strategyproofness guarantees.

We start with all hospitals having an empty bundle, i.e., $X_h = \emptyset$ for all $h \in H$. We order the doctors from d_1 to d_m . Each doctor d_i goes down their ranking list, and *proposes* to hospitals in decreasing order of preferences. When d_i proposes to the hospital h , if $\Delta_h(X_h, d_i) = 1$, then the hospital *accepts*. Else the hospital *rejects* and the doctor moves on to the next hospital.

ALGORITHM 2: Doctor Serial Dictatorship

```
 $X = (X_0, X_1, \dots, X_n) \leftarrow (D, \emptyset, \dots, \emptyset)$ 
for  $i$  in 1 to  $m$  do
  for  $j$  in 1 to  $n$  do
    Let  $h$  be the  $j$ -th highest ranked hospital for doctor  $d_i$ 
    // Doctor  $d_i$  proposes to hospital  $h$ 
    if  $\Delta_h(X_h, d_i) = 1$  then
      // Hospital  $h$  accepts
       $X_h \leftarrow X_h + d_i$ 
       $X_0 \leftarrow X_0 \setminus d_i$ 
      Break
    if  $\Delta_h(X_h, d_i) = 0$  then
      // Hospital  $h$  rejects
      Continue
  return  $X$ 
```

Theorem 3.3. *The doctor serial dictatorship mechanism has the following properties:*

- (i) *The mechanism is doctor strategyproof*
- (ii) *The output allocation is stable*
- (iii) *The algorithm runs in polynomial time*

- (iv) *The output allocation is 2-approximately max hospital-USW.*
- (v) *The mechanism is 2-approximately hospital strategyproof.*

This strategyproofness analysis is tight, once again due to Example 2.7. The same misreport described in the example shows that we cannot beat 2-approximate hospital-SP with the serial dictatorship mechanism.

4 Doctors with Cardinal Utilities

We now turn our attention to a more general problem where hospitals still have matroid rank valuations but doctors have cardinal valuations over the doctors. These valuations are denoted by the function $c_d : H \rightarrow \mathbb{R}$; $c_d(h)$ denotes the utility that doctor d derives from being assigned to the hospital h . Note that cardinal preferences can encode both ties and incomplete orders in doctors preferences; we assume setting $c_d(h) < 0$ implies that the doctor d prefers to be unallocated than allocated to hospital h .

In this section, we continue to define stability using Definition 2.4. Specifically, we note that Definition 2.3 continues to be well-defined and Observation 2.5 holds in the presence of ties.

4.1 Maximizing Doctor Welfare

We first show that the problem of maximizing doctor-USW is an instance of the weighted matroid intersection problem which admits polynomial time algorithms. The maximum doctor-USW allocation among all non-redundant allocations is guaranteed to be stable, so this offers us an efficient algorithm to compute a stable, doctor-USW maximizing allocation. We can in fact make a slightly stronger statement (proof in Appendix D).

Lemma 4.1. *There exists a polynomial time algorithm that computes a stable, max doctor-USW allocation. Moreover, the output allocation has the highest possible hospital-USW among all stable, max doctor-USW allocations.*

By setting $c_d(h)$ values appropriately, the above result can be used to compute both the outputs of Algorithms 1 and 2. If doctors have ordinal preferences, we use $u_d(h)$ to denote the Borda score of hospital h according to doctor d , i.e., if d ranks h at the k -th position, then $u_d(h) = n - k$. If we instantiate Lemma 4.1 with $c_d(h) = (2n + 1)^{n-d} u_d(h)$, we recover the output of the doctor Serial Dictatorship mechanism (Algorithm 2). If we set $c_d(h) = M + (2n + 1)^{n-d} u_d(h)$, where M is a very large number, we recover Algorithm 1.

Assuming we receive cardinal valuations for each doctor c_d , we can compute a doctor-USW maximizing non-redundant allocation. Moreover, since each doctor gets allocated to at most one hospital, by instantiating Lemma 4.1 with valuations $c'_d(h) = \log c_d(h)$, we compute a non-redundant doctor-NSW maximizing allocation. Both these allocations are stable because, if either allocation is not stable, there is a hospital h and a doctor d such that we can move the doctor d to hospital h to increase its utility while not affecting any other doctor's utilities (Observation 2.5). This contradicts the fact that we picked the doctor-USW (or doctor-NSW) maximizing allocation.

4.2 Doctor-Optimal and Hospital Nash Optimal Allocations

In Lemma 4.1, we showed that subject to maximizing doctor-USW, we can maximize hospital-USW as well. In this section we show that subject to maximizing doctor-USW, we can maximize hospital-NSW. By appropriately setting doctor valuations, we can prove a similar result with doctor-NSW maximizing allocations.

ALGORITHM 3: Maximizing Hospital Nash Welfare subject to Maximizing Doctor Welfare

```
for  $k$  in 1 to  $m$  do
   $X = (X_0, X_1, \dots, X_n) \leftarrow$  a  $k$ -MDW allocation
   $c_i \leftarrow |X_i|$  for each  $i \in H$ 
  while 1 do
    for  $i$  in 1 to  $n$  do
      for  $j$  in 1 to  $n$  do
        if  $|X_i| \geq |X_j| + 2$  or  $|X_i| = |X_j| + 1$  and  $j < i$  then
          
$$c'_p = \begin{cases} c_i - 1 & p = i \\ c_j + 1 & p = j \\ c_p & p \neq i, j \end{cases}$$

           $X' \leftarrow$  a  $k$ -MDW allocation with respect to the capped valuations
          if  $X'$  is a  $k$ -MDW allocation with respect to the original valuations then
             $X \leftarrow X'$ 
             $\vec{c} \leftarrow \vec{c}'$ 
        if  $X = X'$  then
          Set  $X^k \leftarrow X$ 
          Break out of the while loop
```

Return the allocation X^k with the highest doctor-USW and subject to that, the one that maximizes $\prod_{h \in H} (v_h(X_h^k))$

We say that an allocation X is k -maximal doctor-USW (k -MDW) if it maximizes doctor-USW, subject to hospitals having a USW of exactly k . Such an allocation can be computed in polynomial time (using a variant of Lemma 4.1).

Our algorithm is based on *utility capping* and *local search*. Given a hospital $h \in H$ with a valuation function v_h and an integer $c > 0$, we define the *capped valuation function* of h as $v_h^c(S) = \min\{v_h(S), c\}$. We note that if v_h is a matroid rank function, so is v_h^c . The valuation v_h^c limits hospital h to a maximum of c doctors.

Our algorithm works as follows: for each value of k , we find a k -MDW allocation X . Next, we cap the utility of every hospital h at $c_h = |X_h|$.

We utilize a local search technique to find a doctor-USW optimal/hospital-NSW optimal allocation, subject to maintaining total hospital welfare at k . We identify two hospitals i and j such that $|X_i| \geq |X_j| + 2$ or $|X_i| = |X_j| + 1$ and $j < i$, and there is a k -MDW allocation X' such that

- (i) $|X'_i| = |X_i| - 1$,
- (ii) $|X'_j| = |X_j| + 1$, and
- (iii) $|X'_p| = |X_p|$ for all $p \neq i, j$.

We can efficiently check whether such an allocation X' exists by increasing the cap of hospital j , c_j , by 1 and decreasing the cap of hospital i , c_i , by 1 before computing a k -MDW allocation. We check if this cap change reduces the doctors' welfare; if so, we reject the new allocation and try a different pair of candidate hospitals $i, j \in H$.

If there exists an allocation X' where hospitals i and j 's caps can be adjusted, we replace X with X' and repeat the process. We stop when an update is no longer possible. We do this for each value of $k \in [m]$ to generate m allocations $X^1, \dots, X^m$; we then simply pick one that maximizes doctor-USW, and subject to that maximizes the hospital-NSW. The steps are described in Algorithm 3.

The proof of Algorithm 3's correctness (Theorem 4.3) relies on the following key lemma (proof in

Appendix D.1).

Lemma 4.2. *Let X and Y be two k -MDW allocations. Then, for any hospital $i \in H$ such that $|X_i| < |Y_i|$, there exists a hospital $j \in H$ such that $|X_j| > |Y_j|$ and a k -MDW allocation X' such that*

- (i) $|X'_i| = |X_i| + 1$,
- (ii) $|X'_j| = |X_j| - 1$, and
- (iii) $|X'_p| = |X_p|$ for all $p \neq i, j$.

This lemma almost immediately gives us the main proof of this section (proof in Appendix D).

Theorem 4.3. *Algorithm 3 terminates in polynomial time and outputs an allocation X that maximizes doctor welfare and subject to that maximizes hospital Nash welfare.*

When the allocation that maximizes doctor-USW is unique, then the above theorem is not meaningful. However, when there are many ties in the optimal doctor-USW allocation, the above theorem suggests that we can find a fair one.

4.3 Hardness of Maximizing Hospital-NSW subject to Stability

We show that finding an approximately max hospital-NSW allocation subject to stability is computationally intractable. The statement we show is in fact stronger than that (proof in Appendix D):

Theorem 4.4. *The problem of deciding whether there exists a stable allocation where all hospitals are allocated at least one doctor is NP-complete when hospitals have capped binary additive valuations.*

This result implies the intractability of other forms of individual fairness. For example, it implies that computing a stable egalitarian allocation, i.e., one that maximizes $\min_h v_h(X_h)$ subject to stability, is intractable.

5 Conclusions and Future Work

In this work we offer a comprehensive analysis of two-sided matchings where one side has MRF valuations over the other. We offer both sequential and optimization based approaches to computing stable, approximately efficient and approximately strategyproof allocations. The problem of computing stable, hospital strategyproof allocations remains open. Expanding these results to superclasses of MRF valuations is an important direction as well.

References

- [1] Haris Aziz, Md. Shahidul Islam, and Szilvia Pápai. Strategyproof maximum matching under dichotomous agent preferences, 2025. URL <https://arxiv.org/abs/2502.09962>.
- [2] Moshe Babaioff, Tomer Ezra, and Uriel Feige. Fair and truthful mechanisms for dichotomous valuations. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, pages 5119–5126, 2021.
- [3] Siddharth Barman and Paritosh Verma. Existence and computation of maximin fair allocations under matroid-rank valuations. In *Proceedings of the 20th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 169–177, 2021.
- [4] Nawal Benabbou, Mithun Chakraborty, Ayumi Igarashi, and Yair Zick. Finding fair and efficient allocations when valuations don’t add up. In *Proceedings of the 13th International Symposium on Algorithmic Game Theory (SAGT)*, pages 32–46, 2020.

- [5] Kristóf Bérczi, Naonori Kakimura, and Yusuke Kobayashi. Market pricing for matroid rank valuations. *SIAM Journal on Discrete Mathematics*, 35(4):2662–2678, 2021.
- [6] Piotr Berman, Marek Karpinski, and Alexander Scott. Approximation hardness of short symmetric instances of MAX-3SAT. *Electronic Colloquium on Computational Complexity*, TR03, 2003.
- [7] Carl Brezovec, Gérard Cornuéjols, and Fred Glover. Two algorithms for weighted matroid intersection. *Math. Program.*, 36(1):39–53, October 1986.
- [8] Deeparnab Chakrabarty, Yin Tat Lee, Aaron Sidford, Sahil Singla, and Sam Chiu-wai Wong. Faster matroid intersection. In *Proceedings of the 60th Symposium on Foundations of Computer Science (FOCS)*, pages 1146–1168, 2019.
- [9] Cyrus Cousins, Vignesh Viswanathan, and Yair Zick. The good, the bad and the submodular: Fairly allocating mixed manna under order-neutral submodular preferences. In *Proceedings of the 19th International Conference on Web and Internet Economics (WINE)*, pages 207–224, 2023.
- [10] Gergely Csáji, Tamás Király, and Yu Yokoi. Approximation algorithms for matroidal and cardinal generalizations of stable matching. In *Proceedings of the 2023 Symposium on Simplicity in Algorithms (SOSA)*, pages 103–113, 2023.
- [11] Jack Edmonds. Matroid intersection. *Annals of Discrete Mathematics*, 4:39–49, 1979.
- [12] Tamás Fleiner. A matroid generalization of the stable matching polytope. In *Proceedings of the 8th Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 105–114, 2001.
- [13] Tamás Fleiner and Naoyuki Kamiyama. A matroid approach to stable matchings with lower quotas. *Mathematics of Operations Research*, 41(2):734–744, 2016.
- [14] David Gale and Lloyd S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.
- [15] Magnús M. Halldórsson, Kazuo Iwama, Shuichi Miyazaki, and Hiroki Yanagisawa. Improved approximation results for the stable marriage problem. *ACM Trans. Algorithms*, 3(3):30–es, August 2007.
- [16] John William Hatfield and Paul R Milgrom. Matching with contracts. *American Economic Review*, 95(4):913–935, 2005.
- [17] Chien-Chung Huang. Classified stable matching. In *Proceedings of the 2010 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1235–1253, 2010.
- [18] Robert W. Irving. Stable marriage and indifference. *Discrete Applied Mathematics*, 48(3):261–272, 1994.
- [19] Kazuo Iwama, David Manlove, Shuichi Miyazaki, and Yasufumi Morita. Stable marriage with incomplete lists and ties. In *Proceedings of the 26th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 443–452, 1999. ISBN 978-3-540-48523-0.
- [20] Naoyuki Kamiyama. A matroid generalization of the super-stable matching problem. *SIAM Journal on Discrete Mathematics*, 36(2):1467–1482, 2022.
- [21] Alexander S. Kelso, Jr., and Vincent P. Crawford. Job matching, coalition formation, and gross substitutes. *Econometrica*, 50(6):1483–1504, 1982.
- [22] Elsie Marie Knoer and Vincent P. Crawford. Job matching with heterogeneous firms and workers. *Econometrica*, 49(2):437–450, 1981.
- [23] Donald E. Knuth. *Stable Marriage and Its Relation to Other Combinatorial Problems: An Introduction to the Mathematical Analysis of Algorithms*, volume 10. American Mathematical Society and Centre de Recherches Mathématiques, 1997.
- [24] Benny Lehmann, Daniel Lehmann, and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55(2):270–296, 2006.
- [25] David F. Manlove. The structure of stable marriage with indifference. *Discrete Applied Mathematics*, 122(1):167–181, 2002.
- [26] David F Manlove, Robert W Irving, Kazuo Iwama, Shuichi Miyazaki, and Yasufumi Morita. Hard variants of stable marriage. *Theoretical Computer Science*, 276(1):261–279, 2002.
- [27] Eric McDermid. A $3/2$ -approximation algorithm for general stable marriage. In *Proceedings of the 2009 International Colloquium on Automata, Languages and Programming (ICALP)*, pages 689–700,

2009. ISBN 978-3-642-02927-1.
- [28] James Oxley. *Matroid Theory*. Oxford Graduate Texts in Mathematics. Oxford University Press, 2nd edition, 2011.
 - [29] Renato Paes-Leme. Gross substitutability: An algorithmic survey. *Games and Economic Behavior*, 106:294–316, 11 2017.
 - [30] Katarzyna Paluch. Faster and simpler approximation of stable matchings. *Algorithms*, 7(2):189–202, 2014.
 - [31] Alvin Roth and Elliott Peranson. The redesign of the matching market for american physicians: Some engineering aspects of economic design. *American Economic Review*, 89(4):748–780, 1999.
 - [32] Alvin E. Roth. The evolution of the labor market for medical interns and residents: A case study in game theory. *Journal of Political Economy*, 92(6):991–1016, 1984.
 - [33] Alvin E. Roth. Stability and polarization of interests in job matching. *Econometrica*, 52(1):47–58, 1984.
 - [34] Alvin E Roth. The college admissions problem is not equivalent to the marriage problem. *Journal of Economic Theory*, 36(2):277–288, 1985.
 - [35] Alvin E. Roth. Conflict and coincidence of interest in job matching: Some new results and open questions. *Mathematics of Operations Research*, 10(3):379–389, 1985.
 - [36] Tayfun Sönmez. Manipulation via capacities in two-sided matching markets. *Journal of Economic theory*, 77(1):197–204, 1997.
 - [37] Tayfun Sönmez. Strategy-proofness in many-to-one matching problems. *Economic Design*, 1: 365–380, 2004.
 - [38] Thorben Troebst and Vijay V. Vazirani. Cardinal-utility matching markets: The quest for envy-freeness, pareto-optimality, and efficient computability. In *Proceedings of the 25th ACM Conference on Economics and Computation (EC)*, page 42, 2024.
 - [39] Vignesh Viswanathan and Yair Zick. A general framework for fair allocation under matroid rank valuations. In *Proceedings of the 24th ACM Conference on Economics and Computation (EC)*, pages 1129–1152, 2023.
 - [40] Yu Yokoi. A generalized polymatroid approach to stable matchings with lower quotas. *Mathematics of Operations Research*, 42(1):238–255, 2017.

A Missing Proofs from Section 2

Observation 2.5. Given a non-redundant allocation X , a blocking pair exists if and only if there exists some doctor d and hospital h such that $\Delta_h(X_h, d) = 1$ and $h \succ_d X(d)$.

Proof. ($\implies$) If a blocking pair (h, S) exists, then since $v_h(S) > v_h(X_h)$, there is some doctor $d \in S$ such that $\Delta_h(X_h, d) = 1$; this follows from the matroid augmentation property (Observation 2.1). For this doctor $d \in S$, from the definition of a blocking pair, $h \succ_d X(d)$.

($\impliedby$) if there exists some doctor d and hospital h such that $\Delta_h(X_h, d) = 1$ and $h \succ_d X(d)$, then it is easy to see that $(h, X_h + d)$ forms a blocking pair: $v_h(X_h + d) > v_h(X_h)$, and for all $d' \in X_h + d$ $h \succeq_{d'} X(d')$. $\square$

Lemma 2.6. *When hospitals have matroid rank valuations, every stable allocation is a 2-approximation of the maximum hospital-USW.*

Proof. Since doctors have a strict and complete preference ordering over hospitals, for an allocation to be stable, it has to be maximal — no unmatched doctor can have a non-zero marginal value for any hospital given the current assignment. It is well known that when hospitals have submodular valuations, maximal allocations give a 2-approximation to the optimal hospital-USW [24]. $\square$

B Missing Proofs for the HWSD Mechanism

Since the hospital-SP guarantees are the most non-trivial, we first show all the other guarantees, relegating the hospital-SP guarantees to Section B.1.

Theorem B.1. *The high welfare Serial Dictatorship (HWSD) mechanism has the following properties:*

- (i) *The mechanism is doctor-SP.*
- (ii) *The algorithm runs in polynomial time.*
- (iii) *The output allocation maximizes hospital-USW.*
- (iv) *The output allocation is stable.*

Proof. We divide this proof into a series of claims.

Claim B.2. *Algorithm 1 outputs an allocation that maximizes hospital-USW.*

Proof. At every round, we maintain the invariant that there exists a socially optimal allocation Y so that each hospital h 's assigned bundle of doctors X_h is contained in Y_h . Thus, when the algorithm terminates, no additional doctors can be assigned, and the resulting allocation maximizes hospital-USW. $\square$

Claim B.3. *The mechanism is doctor-SP.*

Proof. The mechanism is doctor-SP since each doctor d 's preferences are only used to assign them to their highest ranked hospital that maintains non-redundancy and satisfies the max USW invariant. Since the max USW invariant and non-redundancy do not depend on the doctor d 's preferences, misreporting preferences can only lead to a worse assignment for d (and any doctor by extension). $\square$

Claim B.4. *Algorithm 1 computes a stable allocation.*

Proof. A new doctor only gets added to a hospital's allocation if it does not violate non-redundancy. Therefore, the final allocation must be non-redundant.

Assume for contradiction that the output allocation (say X) has a blocking pair. By Observation 2.5, there must be some doctor d and some hospital h such that $\Delta_h(X_h, d) = 1$ and $h \succ_d X(d)$. If d is not allocated in X , then we get that d can be added to X_h , and the social welfare strictly increases, contradicting the fact that the algorithm finds a hospital-USW maximizing allocation. Therefore, assume that $h_d = X(d)$ is the hospital d was assigned by Algorithm 1. Suppose that we move the doctor d from X_{h_d} to X_h . This creates a new non-redundant allocation X' where $X'_{h_d} = X_{h_d} - d$ and $X'_h = X_h + d$, and the remaining bundles are the same. Since the output of Algorithm 1 is non-redundant and $\Delta_h(X_h, d) = 1$, we have that X' is also hospital-USW maximizing: hospital h_d lost a utility point and hospital h gained one.

Let X^t be the interim allocation at the beginning of the t -th iteration of Algorithm 1. Since doctors are never unassigned from hospitals, we have that for every hospital $h \in H$, $X_h^t \subseteq X_h^{t+1}$. Consider the iteration t where d was allocated and let X^t be the allocation at the start of this iteration. Since X_h^t is a subset of the final allocation X_h to h and $\Delta_h(X_h, d) = 1$, we must have $\Delta_h(X_h^t, d) = 1$ by submodularity. By construction there exists a max hospital-USW allocation X' with $X_{h'} \subseteq X'_{h'}$ for all $h' \neq h$, and $X_h^t + d \subseteq X'_h$. This implies that the hospital h_d allocated by Algorithm 1 must satisfy $h_d \succ_d h$ contradicting the definition of a blocking pair. So, no such blocking pair exists, and the allocation is stable. $\square$

Claim B.5. *Algorithm 1 runs in polynomial time.*

Proof. Algorithm 1 runs a for loop over the doctors and then examines hospitals in decreasing order of doctor preference in another for loop, for a maximum of nm iterations. In each iteration, we test whether there exists some hospital-USW maximizing allocation Y such that $X_h \subseteq Y_h$ for all $h \in H$. Thus, to prove our claim we need to show that it is possible to compute the maximum hospital-USW possible given a partial allocation X . Computing the maximum hospital-USW allocation can be done in polynomial time using value queries, see e.g., Benabbou et al. [4], who reduce the problem to the matroid intersection problem [11].

We also note that for any $T \subseteq D$, if v_h is a matroid rank function, v'_h defined as $v'_h(S) = v_h(T \cup S) - v_h(T)$ is also a matroid rank function. Therefore, we can compute the maximum hospital-USW possible with hospital valuations v'_h defined as $v'_h(S) = v_h(X_h \cup S) - v_h(X_h)$ for each $h \in H$. We can therefore efficiently determine if there exists a max hospital-USW Y such that for each h , $X_h \subseteq Y_h$. We conclude that Algorithm 1 runs in polynomial time. $\square$

$\square$

B.1 Proof of Hospital-SP Guarantees

Our hospital-SP guarantees only apply when hospital valuations are defined by binary OXS functions. This means each hospital's valuations v_h is defined by a bipartite graph G_h with doctors on one side and an arbitrary number of nodes (called *slots*) on the other. The value of any set of doctors $S \subseteq D$ is the maximum cardinality matching in the subgraph of G_h induced by the doctors in S and all the slots sharing edges with S . In other words, the value $v_h(S)$ is the number of slots that can be filled by the set of doctors S , where each doctor d can only take up neighboring slots in the graph G_h .

Algorithm 1 is 2-approximately hospital-SP when hospitals have binary OXS valuations. That is, for any hospital, misreporting their valuation can increase a hospital's bundle size by no more than a factor of 2.

To prove this property, we begin with a few definitions. Given a subset of doctors $T \subseteq D$, we define the valuation f_T as $f_T(S) = |S \cap T|$; that is, each doctor in T adds a value of 1. Given an allocation X and a set of doctors D' , we define the allocation $X \setminus D'$ as the allocation where each hospital h receives the set $X_h \setminus D'$. Similarly, given an allocation X and a set of doctors D' , we define $X \cap D'$ as the allocation where each hospital h receives the set $X_h \cap D'$.

We define the following ordering over allocations: given two non-redundant allocations X and Y , we say that $X \succ Y$ if (a) $\text{USW}(X) > \text{USW}(Y)$, or (b) $\text{USW}(X) = \text{USW}(Y)$, and there is some doctor d_j such that $X(d_j) \succ_{d_j} Y(d_j)$ and for all doctors d_k where $k < j$ we have $X(d_k) = Y(d_k)$. We call this ordering the *HWSD ordering*. Let X be the output of Algorithm 1; then for any other non-redundant allocation Y , we must have $X \succ Y$. Since this observation is used several times in our proofs, we state it formally.

Observation B.6. Let X be the output of Algorithm 1, and let X' be any other non-redundant allocation. Then $X \succ X'$ according to the HWSD ordering.

To prove our result, we first show that it suffices to consider simple misreports, of the form f_T , when considering hospital deviations. Then we show that the outcome from reporting f_T and the outcome from reporting v_h cannot differ significantly in size. While the proof follows the broad structure of Babaioff et al. [2], our proofs of Lemma B.7 and B.8 are more technically involved due to the two-sided nature of our problem.

Lemma B.7. Let X be the output of Algorithm 1. If hospital h reports f_T instead of v_h where $T \subseteq X_h$ and all the other hospitals report the same valuations, then Algorithm 1 outputs an allocation Y where $Y_h = T$.

Proof. When representing the valuation function f_T , we assume the graph G_{f_T} used is a subgraph of G_h (used for representing v_h). Indeed, it is always possible to represent f_T using an appropriate subgraph of G_h by keeping the edges of the bipartite graph representing G_h that are in a maximum cardinality matching of doctors in T to slots. For both allocations X and Y , fix a matching of the allocated bundle for each hospital to the slots in their valuation graph; specifically for the hospital h , ensure the matching under f_T is a subset of the matching under v_h . Essentially, this means each doctor is not only associated with a hospital but also with a specific slot in the hospital's preference graph G_h . To make notation easier, we call the preference profile where every hospital h reports v_h as the *old* preferences, and the profile where h reports f_T the *new* preferences.

The proof is trivial for the case when $T = X_h$, since at each iteration, the algorithm makes the exact same decision under both profiles. It suffices to consider the case where $|T| = |X_h| - 1$, since we can then inductively assume $v_h = f_T$ and apply the same argument to $|T'| = |T| - 1$ to obtain an arbitrary subset of X_h .

Let d be the doctor in $X_h \setminus T$. Assume for contradiction that $|Y_h| < |T|$. We create an allocation X' from X by moving d from hospital h to the hospital it is allocated to in Y – hospital j where d is matched to some slot q . If slot q is empty in the matching of allocation X_j in the graph G_j , we add d to slot q of hospital j and we stop. Otherwise, we swap d with d' in slot q of X_j . We then repeat this process with d' . This transfer creates an alternating path of doctors and (hospital, slot) pairs; we call this path P (see Figure 1 for an illustration). This path is finite because after every move, we strictly increase the number of (hospital, slot) pairs which have the same doctor allocated in both X' and Y . We denote this path as $((h_1, q_1), d_1, (h_2, q_2), d_2, \dots, d_k, (h_{k+1}, q_{k+1}))$; the interpretation being that the doctor d_i was moved from the hospital slot pair (h_i, q_i) to (h_{i+1}, q_{i+1}) to create X' . The resulting assignment X' is non-redundant with respect to the new preferences. Moreover, $X' \neq Y$ since $X' = Y$ implies that $|Y_h| = |X'_h| = |T|$ which implies the lemma is satisfied. Therefore the allocation X' must be strictly worse according to the HWSD ordering than Y with respect to the new preferences (Observation B.6). Let the doctors involved in the path P be denoted by D_P . The set of doctors D_P

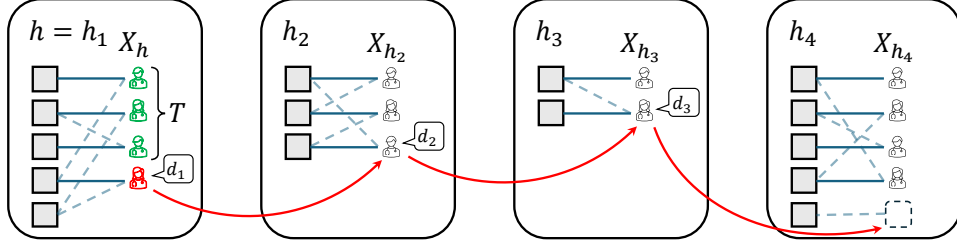


Figure 1: Illustration of the path transfer executed in the proof of Lemma B.7. Each hospital has availability slots (gray squares) that can be matched to different doctors (denoted by blue edges). Full blue edges denote the actual matching of doctors to slots, and dashed edges denote possible alternative slots. The doctor d_1 is assigned to h under X_h , but not under Y (where h receives T). We transfer d_1 to the slot+hospital it is assigned to under Y . That slot is occupied by the doctor d_2 who is moved to the spot of d_3 . Finally, d_3 is assigned to their slot with h_4 under Y , which is unoccupied under X . This same path is reversed later in the proof to create the allocation Y' from the allocation Y .

have the same allocation in both X' and Y . Therefore, if X' is worse than Y , it must be because of the doctors not present in the path. That is, the allocation $X' \setminus D_P$ is worse according to the HWSD ordering than $Y \setminus D_P$. Since this set of doctors has the same allocation in both X and X' , we have that $X \setminus D_P$ is worse than $Y \setminus D_P$ according to the HWSD ordering.

Now, we reverse the path and apply it to Y to get allocation Y' which is non-redundant with respect to the old preferences. By reversing the path, we mean the following: if the path is $((h_1, q_1), d_1, (h_2, q_2), d_2, \dots, d_k, (h_{k+1}, d_{k+1}))$, we move d_j from the hospital slot pair (h_{j+1}, d_{j+1}) to (h_j, d_j) to create Y' ; note that this operation is well-defined. Reversing the path P results in a non-redundant allocation (with respect to the old preferences) if no two doctors are matched to the same slot for some hospital. This happens because at each step of this reversed path we either

- (a) allocate d to some slot in Y'_h ,
- (b) swap a doctor allocated to some slot in a hospital with another doctor, or
- (c) remove a doctor from a hospital.

The second and third operations clearly preserve non-redundancy with respect to both the old and new preferences. The first operation preserves non-redundancy since we assume the matching of Y_h in G_h is a subset of the matching of X_h in G_h . So, if the doctor d was allocated to slot q in X_h , then this slot must remain unoccupied in Y_h .

We observe that $Y' \neq X$ since $|Y'_h| \leq |T| < |X_h|$. Therefore, Y' must be strictly worse than X according to the red jacket objective which implies that doctors not present in the path are worse off in Y than in X according to the red jacket objective; that is, the allocation $Y \setminus D_P$ is worse than the allocation $X \setminus D_P$ according to the HWSD ordering (Observation B.6). This gives us a contradiction. $\square$

Our next lemma shows that being honest does not significantly reduce the size of your bundle.

Lemma B.8. *Consider some hospital h . Fixing the reports of all other hospitals, let X and Y be the resulting allocation when h reports f_T (for some T) and v_h , respectively. If $v_h(T) = |T|$ and $X_h = T$, then $|Y_h| \geq \frac{|T|}{2}$.*

Proof. We call the preference profile where h reports f_T and receives $X_h = T$ the *misreported* preferences. For both allocations, fix a matching of the allocated bundle for each hospital to the slots in their valuation graph. Like the proof of Lemma B.7, when representing the valuation function f_T , we assume the graph G_{f_T} used is a subgraph of G_h (used for representing v_h).

We also assume without loss of generality that the matching for the bundles X_h and Y_h is chosen in a manner that maximizes the number of slots that receive the same doctor under both matchings.

Suppose that there are k slots filled in X_h which are not filled in Y_h , with doctors $\{d_1, \dots, d_k\}$ assigned to them. For each of these k doctors we create a path as we did in Lemma B.7: we move the doctor d_j to the hospital-slot pair it is assigned to in Y ; if that slot is empty then we stop, otherwise we transfer the doctor in that slot to the slot they were assigned to under Y . Doing this, we create k paths $P^{(1)}, \dots, P^{(k)}$. These paths are disjoint, since we fixed the matching of doctors to slots for each hospital in both X and Y .

Consider a path $P^{(j)}$ starting with a doctor d_j . If the path $P^{(j)}$ only involves slots of the hospital h , then transferring doctors along this path (as we did in Lemma B.7) increases the number of slots in G_h which receive the same allocation in both X_h and Y_h , without changing the hospital h 's utility. This contradicts the assumption that the matchings we chose for h under X_h and Y_h maximally intersect. Therefore, all paths must contain one hospital-slot pair that is not the hospital h .

Assume that transferring doctors along the path $P^{(j)}$ creates a non-redundant allocation X' with respect to the misreported preferences (when hospital h reported f_T). Using Observation B.6, it must be the case that X' is worse than X according to the HWSD ordering. Since only doctors on the path $P^{(j)}$ had their assignment changed, this means $X' \cap D_{P^{(j)}}$ is worse than $X \cap D_{P^{(j)}}$ where $D_{P^{(j)}}$ is the set of doctors in the path $P^{(j)}$. This implies $Y \cap D_{P^{(j)}}$ is worse than $X \cap D_{P^{(j)}}$. If this is true, we can reverse the path and apply it to Y to improve the allocation according to the HWSD ordering, which is a contradiction. This follows from arguments similar to the previous proof.

Therefore, for all of the paths, transferring along these paths creates an allocation X' that is *not non-redundant* with respect to the misreported preferences. The only way this can happen is if we move a doctor $d \in Y_h \setminus T$ to X'_h ; this follows from the fact that the only hospital with a different set of preferences in both allocations is hospital h . Therefore, each of these paths has a doctor from $Y_h \setminus X_h$ and since the paths are disjoint, no two paths share a doctor.

Therefore, we have $|Y_h \setminus X_h| \geq k \geq |X_h| - |Y_h|$ or that $|Y_h| \geq |X_h| - |Y_h \setminus X_h| \geq |X_h| - |Y_h|$, which implies that $|Y_h| \geq \frac{|X_h|}{2}$. $\square$

Combining Lemmas B.7 and B.8 proves our claim.

Theorem B.9. *Algorithm 1 is 2-approximately hospital-strategyproof when hospitals have binary OXS valuations.*

Proof. Suppose that when hospital h reports their valuation v_h truthfully, the resulting output of Algorithm 1 is X , and hospital h receives the bundle X_h . Suppose that hospital h reports the valuation v'_h instead of v_h , and Algorithm 1 outputs the allocation Y . Let $T \subseteq Y_h$ be a subset of doctors such that $|T| = v_h(T) = v_h(Y_h)$; intuitively, T is the set of doctors in Y_h that hospital h actually wants to receive. Suppose that h reports f_T instead of v'_h , and let the resulting allocation be Z . From Lemma B.7 we know that $v_h(Z_h) = |T| = v_h(Y_h)$. If h reports v_h and receives the bundle X_h , then from Lemma B.8 we get that

$$v_h(X_h) \geq \frac{1}{2}v_h(Z_h) \geq \frac{1}{2}v_h(Y_h).$$

Therefore, Algorithm 1 is 2-approximately hospital strategyproof. $\square$

This guarantee can be strengthened when hospitals have binary capped additive valuations.

Lemma 3.2. *When hospitals have binary capped additive valuations, the HWSD mechanism is hospital-SP.*

Proof. To prove this, we prove the following improved version of Lemma B.8.

Lemma B.10. *Consider some hospital h . Fixing the reports of all other hospitals, let X and Y be the resulting allocation when h reports f_T and v_h , respectively. When agents report capped additive valuations, if $v_h(T) = |T|$ and $X_h = T$, then $|Y_h| \geq |T|$.*

Proof. Similar to Lemma B.8, we call the preference profile where h reports f_T the *misreported preferences*, and the profile where h reports f_T the *true preferences*. Assume for contradiction that $|Y_h| < |T|$. Since the allocation X is non-redundant with respect to the true preferences, it must be worse according to the HWSO ordering than Y . This implies $USW(Y) \geq USW(X)$.

Therefore, since $|Y_h| < |T| = |X_h|$, there must be some hospital h' such that $|Y_{h'}| > |X_{h'}|$. Let d be some doctor in $Y_{h'} \setminus X_{h'}$. We create a path starting from the doctor d just like we have done in the previous lemmas. However, since all agents have capped additive valuations, we can be less rigorous with how we define these paths. We move the doctor d to the hospital h_1 it is allocated to in the allocation X . If this transfer results in h_1 's bundle exceeding its capacity, then there must be some doctor d_1 in Y_{h_1} which is allocated to some other hospital (say h_2) in X . We repeat this process with d_1 till we make a transfer that respects the capacity constraints of each hospital. This creates a path $P = (h_0 = h', d_1 = d, h_1, d_2, \dots, d_k, h_{k+1})$. We define a transfer along this path as moving the doctor d_i from hospital h_{i-1} to h_i . The transfer along the path P when applied to the allocation Y gives us an allocation Y' which is non-redundant with respect to both the true preferences.

Since $|Y_h| < |X_h|$, the only place that hospital h could appear in this path is right at the end (at h_{k+1}), since an addition of a single doctor to hospital h (under the allocation Y) will not violate its capacity constraints. This means if we reverse the path and apply it to the allocation X , the new allocation (say X') will be non-redundant with respect to the misreported preferences.

Since X and X' are non-redundant with respect to the misreported preferences X must be better than X' according to the HWSO ordering (Observation B.6). Since the only difference between the two allocations are the agents on the path, the allocation $X' \cap D_P$ is worse than the allocation $X \cap D_P$ according to the HWSO ordering. We now observe that $X' \cap D_P$ is equal to $Y \cap D_P$ and $X \cap D_P$ is equal to $Y' \cap D_P$. This implies $Y \cap D_P$ is worse than $Y' \cap D_P$ according to the HWSO ordering. Since the allocations Y and Y' only differ in the assignments of the doctors in D_P , this implies Y is worse than Y' according to the HWSO ordering. This contradicts the fact that Algorithm 1 outputs Y under the true preferences. $\square$

Plugging this improvement into the proof of Theorem B.9 gives us the required result. $\square$

C Missing Proofs for the SD Algorithm

Since the hospital-SP guarantees are the most non-trivial, we first show all the other guarantees, relegating the hospital-SP guarantees to Section C.1.

Theorem C.1. *The doctor serial dictatorship mechanism has the following properties:*

- (i) *The mechanism is doctor strategyproof*
- (ii) *The output allocation is stable*
- (iii) *The algorithm runs in polynomial time*
- (iv) *The output allocation is 2-approximately max hospital-USW.*

Proof. Again, we prove this using a series of claims. Note that (iv) follows from Lemma 2.6.

Claim C.2. *Algorithm 2 runs in polynomial time.*

Proof. Algorithm 2 runs in polynomial time since we simply go down the doctors' preference lists in decreasing order, and run a poly-time procedure to check whether each hospital is willing to accept the doctor. $\square$

Claim C.3. *Algorithm 2 outputs a stable allocation.*

Proof. A new doctor d only gets added to a bundle X_h if $\Delta_h(X_h, d) = 1$. Therefore the final allocation is non-redundant.

If there is a blocking pair, there must be some d and h such that $\Delta_h(X_h, d) = 1$ and $h \succ_d X(d)$. Assuming for contradiction that such a pair exists. Consider the iteration where d was allocated and let Y be the allocation at the start of this iteration. By submodularity, we must have $\Delta_h(Y_h, d) = 1$ as well which implies that $X(d) \succ_d h$ contradicting the definition of a blocking pair. So, no such blocking pair exists, and the allocation is stable. $\square$

Claim C.4. *Algorithm 2 is doctor-SP.*

Proof. The mechanism is doctor-SP since each doctor d has a fixed iteration where they are assigned, and at this iteration they are assigned to their most preferred hospital who can accept them. As the hospitals' allocations at the iteration where the doctor d is allocated is independent of the doctor's preferences, the doctor d cannot obtain a strictly better hospital by misreporting their preferences. $\square$

$\square$

C.1 Proof of Hospital-SP Guarantees

In this section, we show that Algorithm 2 is 2-approximately hospital-SP. We fix a hospital h . In this proof, we will also assume for ease of analysis that the unallocated doctors go to hospital h_0 who has the valuation function $v_0(S) = |S|$.

Assume the allocation when all hospitals report truthfully is X . Assume that the hospital h now misreports and let the resulting allocation be Y . In order to prove our claim, we only need to show that $v_h(Y_h) \leq 2|X_h| = 2v_h(X_h)$.

We distinguish between two types of changes to the allocation between X and Y . Some changes are due to doctors not proposing to hospitals that they originally proposed to, and some changes are due to doctors being rejected by hospitals that previously accepted them.

We say that a hospital $h' \in H$ was *denied* a doctor $d \in D$ if d was allocated to h' in X , but the doctor d was allocated to a hospital in Y that d prefers to h' . The set of doctors that h' was denied is denoted $D_{h'}$, and the set of doctors accepted by h' in the generation of Y that were denied from some other hospital is $D_{h'}^a$. Intuitively, $D_{h'}$ is the set of doctors that would have proposed to h' if h had truthfully reported its preferences but now did not propose to h' , and $D_{h'}^a$ is the set of doctors that h' was able to 'steal' from other hospitals due to the misreport by h . Since doctors go down their preference lists when proposing to hospitals, every doctor $d \in D_{h'}^a$ prefers h' to the hospital they were assigned to under X .

We also say that a doctor d was *newly rejected* by hospital h' if d was allocated to h' in X but d was rejected by h' in the generation of Y . Let $R_{h'}$ be the set of doctors newly rejected by the hospital h' , and $R_{h'}^a$ be the set of doctors accepted by h' under Y that were newly rejected by some other hospital.

In what follows, we upper bound $|D_h^a|$ and $|R_h^a|$. This will almost immediately imply our desired upper bound. We first upper bound $|D_h^a|$.

Lemma C.5. $v_h(D_h^a \cup (X_h \cap Y_h)) \leq |X_h|$.

Proof. By definition, all doctors in D_h^a propose to h in the generation of X , and were rejected by h ; this is again because doctors propose to hospitals in decreasing order of preference. Recall that hospitals have matroid rank valuations, therefore, by the matroid augmentation property (Observation 2.1), if $v_h(S) < v_h(T)$ there is some doctor $d \in T \setminus S$ such that $\Delta_h(S, d) > 0$. Therefore, if $v_h(D_h^a \cup (X_h \cap Y_h)) > |X_h| = v_h(X_h)$, there is some doctor d in D_h^a such that $\Delta_h(X_h, d) = 1$. This contradicts the definition of D_h^a since the doctor d would have been accepted by the hospital h during the construction of the allocation X . Therefore, it must be the case that $v_h(D_h^a \cup (Y_h \cap X_h)) \leq |X_h|$. $\square$

Next, we show that $|R_h^a| \leq |X_h \setminus Y_h|$. This proof is somewhat more involved, and requires the construction of an auxiliary set of doctors.

Fix some $h' \in H \setminus \{h\}$. We define $X_{h'}^t$ and $Y_{h'}^t$ as the bundles $X_{h'}$ and $Y_{h'}$ immediately after the iteration where doctor d_t was allocated in Algorithm 2. We construct a new sequence of bundles $W_{h'}^t$ for $t \in [m] \cup \{0\}$. We initialize $W_{h'}^0 = \emptyset$; then, we iterate through t from 1 to m . For each t , we set $W_{h'}^t \leftarrow W_{h'}^{t-1} + d_t$ if:

(R1) $d_t \in X_{h'} \cap Y_{h'}$.

(R2) $d_t \in X_{h'} \setminus Y_{h'}$, and $Y_{h'}^{t-1} + d_t$ is redundant for h' , i.e., $\Delta_{h'}(Y_{h'}^{t-1}, d_t) = 0$.

(R3) $d_t \in D_{h'}^a$.

The final bundle $W_{h'}^m$ contains $X_{h'} \cap Y_{h'}$ and $D_{h'}^a$, as well as any doctors in $X_{h'} \setminus Y_{h'}$ who would not have been selected by h' during the run of Algorithm 2 when h misreports its preferences.

Our basic proof strategy is to show that $W_{h'}^t$ is non-redundant and all doctors in $X_{h'} \setminus W_{h'}^m$ are in $D_{h'}$, i.e., they were denied from h' due to the misreport of hospital h . We use this to obtain the required upper bound on $|R_{h'}|$. This proof will require the following useful matroid property.

Lemma C.6. *Let A and B be two sets of doctors which are non-redundant with respect to some v_h . Let $d \notin A \cup B$ be some doctor such that $\Delta_h(A, d) = 1$ and $\Delta_h(B, d) = 0$. Then there is some doctor $d' \in B \setminus A$ such that $\Delta_h(A, d') = 1$; that is $A + d'$ is non-redundant with respect to v_h .*

Proof. If $|B| > |A|$, then this lemma trivially holds via the matroid augmentation property.

If $|B| \leq |A|$, there must be some doctor $d^* \in (A + d) \setminus B$ such that $\Delta_h(B, d^*) = 1$, by the matroid augmentation property applied to the sets B and $A + d$. Note that $d^* \neq d$ by our assumption in the lemma. Set $B' = B + d^*$ and repeat this process with B' until $|B'| = |A| + 1$.

There must be some element $d' \in B' \setminus A$ such that $\Delta_h(A, d') = 1$, again by invoking the matroid augmentation property with the sets B' and A . This element d' must also be in $B \setminus A$ by the construction of B' . $\square$

We start by showing that $W_{h'}^t$ is non-redundant with respect to the valuation of hospital h' .

Lemma C.7. *At any $h' \in H$ and any $t \in [m] \cup \{0\}$, $W_{h'}^t$ is non-redundant with respect to $v_{h'}$.*

Proof. We prove this by induction. At $t = 0$, $W_{h'}^0 = \emptyset$ and the lemma trivially holds. Assume that $W_{h'}^{t'}$ is non-redundant for all $t' < t$. We show that it holds for t .

We first show that if Rule (R3) is applied, i.e., $d_t \in D_{h'}^a$, then $W_{h'}^t$ is non-redundant. Assume for contradiction that $W_{h'}^{t-1} + d_t$ is redundant. Since $d_t \in D_{h'}^a \subseteq Y_{h'}$ and $Y_{h'}^{t-1} \subseteq Y_{h'}$ we have that $Y_{h'}^{t-1} + d_t$ is non-redundant. Therefore, there must be some doctor $d_j \in W_{h'}^{t-1} \setminus Y_{h'}^{t-1}$ such that $\Delta_{h'}(Y_{h'}^{t-1}, d_j) = 1$ (by Lemma C.6).

Since d_j is not in $Y_{h'}^{t-1}$, and $Y_{h'}^j \subseteq Y_{h'}^{t-1}$, we must have that d_j was not added to $Y_{h'}$ at the j -th iteration and $d_j \notin Y_{h'}$. In particular, since Rules (R1) and (R3) only add doctors that are in $Y_{h'}$, they do not apply

to d_j . The only way that d_j could have been added to $W_{h'}^{j-1}$ is if Rule (R2) applies. Rule (R2) requires that d_j is redundant for $Y_{h'}^{j-1}$. Since $v_{h'}$ is submodular and $Y_{h'}^{j-1} \subseteq Y_{h'}^{t-1}$, $\Delta_{h'}(Y_{h'}^{t-1}, d_j) = 0$ as well. Thus, d_j cannot have been added to $W_{h'}^{j-1}$ and is not in $W_{h'}^{t-1}$, a contradiction. Thus, $W_{h'}^{t-1} + d_t$ is non-redundant if d_t is added via Rule (R3).

Next, suppose that we add some doctor d_t such that $d_t \in X_{h'}$. The only way that this can occur is if d_t is added by applying Rules (R1) and (R2). Assume for contradiction that $W_{h'}^{t-1} + d_t$ is redundant. By Lemma C.6, there must be some doctor $d_j \in W_{h'}^{t-1} \setminus X_{h'}$ such that $j < t$ and $X_{h'} - d_t + d_j$ is non-redundant. The doctor d_j is not in $X_{h'}$ and thus it was not added due to Rules (R1) or (R2). Thus, d_j was added due to Rule (R3) and is in $D_{h'}^a$. Therefore, d_j must have proposed to h' during the generation of allocation X , and was rejected. Recall that $X_{h'}^{j-1} \subseteq X_{h'}$; thus, if $X_{h'} - d_t + d_j$ is non-redundant, then $X_{h'}^{j-1} + d_j$ is non-redundant by submodularity of $v_{h'}$, and h' should have accepted d_j in the generation of X . This is a contradiction and therefore, $W_{h'}^t$ must be non-redundant. $\square$

Lemma C.7 immediately implies an upper bound on the size of $W_{h'}^m$.

Lemma C.8. $|W_{h'}^m| \leq |Y_{h'}|$

Proof. According to Lemma C.7, $W_{h'}^m$ is non-redundant with respect to $v_{h'}$. Therefore, if $|W_{h'}^m| > |Y_{h'}|$, there must be at least one doctor $d \in W_{h'}^m \setminus Y_{h'}$ such that $\Delta_{h'}(Y_{h'}, d) = 1$. This doctor d has to satisfy $d \notin Y_{h'}$. The only way that a doctor not in $Y_{h'}$ is added to $W_{h'}^m$ is via the application of Rule (R2). However, Rule (R2) requires that $\Delta_{h'}(Y_{h'}, d) = 0$; thus, d should have never been added to $W_{h'}^m$ in the first place. $\square$

Lemma C.9. For all $h' \neq h$, $|R_{h'}| \leq |R_{h'}^a|$.

Proof. From the construction of $W_{h'}^m$, every doctor in $X_{h'}$ that is not in $W_{h'}^m$ must be in $D_{h'}$ according to Rule (R2). Therefore, $X_{h'} \setminus W_{h'}^m \subseteq D_{h'}$; in particular, $|X_{h'}| - |X_{h'} \cap W_{h'}^m| \leq |D_{h'}|$. Recall that every doctor in $D_{h'}^a$ is also in $W_{h'}^m$ according to Rule (R3). In other words, $W_{h'}^m$ contains all doctors that were denied from some other hospital when h misreports, and were assigned to h' as a result. Thus, $W_{h'}^m \setminus X_{h'} = D_{h'}^a$; in particular, $|W_{h'}^m| - |X_{h'} \cap W_{h'}^m| = |D_{h'}^a|$.

Finally, by Lemma C.8, $|W_{h'}^m| \leq |Y_{h'}|$. Combining these observations we get:

$$\begin{aligned} |X_{h'}| - |Y_{h'}| &\leq |X_{h'}| - |W_{h'}^m| = |X_{h'}| - |X_{h'} \cap W_{h'}^m| + |X_{h'} \cap W_{h'}^m| - |W_{h'}^m| \\ &= |X_{h'}| - |X_{h'} \cap W_{h'}^m| - (|W_{h'}^m| - |X_{h'} \cap W_{h'}^m|) \leq |D_{h'}| - |D_{h'}^a|. \end{aligned} \quad (1)$$

Thus, $|X_{h'}| - |Y_{h'}| \leq |D_{h'}| - |D_{h'}^a|$. Consider the set of doctors $X_{h'}$ assigned to h' when the hospital h truthfully reports its valuation. It comprises of three disjoint sets of doctors: (a) doctors that are assigned to h' under both $X_{h'}$ and $Y_{h'}$, i.e., $X_{h'} \cap Y_{h'}$ (b) doctors that were denied from h' , i.e. the set $D_{h'}$, and (c) doctors that were newly rejected by h' , i.e. the set $R_{h'}$. Thus, $X_{h'} = (X_{h'} \cap Y_{h'}) \cup D_{h'} \cup R_{h'}$. Similarly, we can show that $Y_{h'} = (X_{h'} \cap Y_{h'}) \cup D_{h'}^a \cup R_{h'}^a$. Therefore,

$$\begin{aligned} |X_{h'}| - |Y_{h'}| &= |X_{h'} \cap Y_{h'}| + |D_{h'}| + |R_{h'}| - (|X_{h'} \cap Y_{h'}| + |D_{h'}^a| + |R_{h'}^a|) \\ &= |D_{h'}| - |D_{h'}^a| + |R_{h'}| - |R_{h'}^a|. \end{aligned} \quad (2)$$

Plugging in the upper bound in Equation (1) into Equation (2) we get

$$|D_{h'}| - |D_{h'}^a| + |R_{h'}| - |R_{h'}^a| \leq |D_{h'}| - |D_{h'}^a| \Rightarrow |R_{h'}| \leq |R_{h'}^a|$$

which concludes the proof. $\square$

We are now ready to prove the final lemma of the theorem.

Lemma C.10. $|R_h^a| \leq |X_h \setminus Y_h|$.

Proof. This proof follows from Lemma C.9. For any $h' \neq h$, we note that each doctor in $R_{h'}^a$ is either from $R_{h''}$ for some h'' or $X_h \setminus Y_h$. Therefore, we have $\bigcup_{h' \neq h} R_{h'}^a \subseteq (\bigcup_{h' \neq h} R_{h'}) \cup (X_h \setminus Y_h)$. By re-arranging terms, we conclude that

$$\left(\sum_{h' \neq h} |R_{h'}^a| \right) - \left| \left(\bigcup_{h' \neq h} R_{h'}^a \right) \cap \left(\bigcup_{h' \neq h} R_{h'} \right) \right| \leq |X_h \setminus Y_h| \quad (3)$$

To prove the lemma, we first use the fact that $R_h^a \subseteq \bigcup_{h' \neq h} R_{h'} \setminus \bigcup_{h' \neq h} R_{h'}^a$. This follows from the definition of R_h^a as doctors which have been newly rejected by other hospitals but accepted by h . We can therefore write,

$$|R_h^a| \leq \sum_{h' \neq h} |R_{h'}| - \left| \left(\bigcup_{h' \neq h} R_{h'}^a \right) \cap \left(\bigcup_{h' \neq h} R_{h'} \right) \right| \leq \sum_{h' \neq h} |R_{h'}^a| - \left| \left(\bigcup_{h' \neq h} R_{h'}^a \right) \cap \left(\bigcup_{h' \neq h} R_{h'} \right) \right| \leq |X_h \setminus Y_h|$$

The second inequality follows from the fact that $|R_{h'}| \leq |R_{h'}^a|$ for all $h' \neq h$, as per Lemma C.9. The third inequality follows from (3). $\square$

Theorem C.11. *Algorithm 2 is 2-approximately strategyproof for the hospitals.*

Proof. Assume that hospital $h \in H$ misreports its valuation, and that the resulting output of Algorithm 2 is now Y instead of X . According to Lemma C.5, $v_h(D_h^a \cup (X_h \cap Y_h)) \leq |X_h|$. According to Lemma C.10, $|R_h^a| \leq |X_h \setminus Y_h|$.

If a hospital h' has a doctor d in $Y_{h'} \setminus X_{h'}$, then d must either be denied from another hospital and assigned to h' , i.e., it is in $R_{h'}^a$; alternatively, d is in $D_{h'}^a$, in which case d prefers h' to whichever hospital they were assigned to under X . We can partition h' 's assignment when it misreports into $Y_{h'} = (Y_{h'} \cap X_{h'}) \cup D_{h'}^a \cup R_{h'}^a$. Using this, we get

$$v_h(Y_h) = v_h((X_h \cap Y_h) \cup D_h^a \cup R_h^a) \leq v_h((X_h \cap Y_h) \cup D_h^a) + |R_h^a| \leq |X_h| + |X_h \setminus Y_h| \leq 2|X_h|.$$

which concludes the proof. $\square$

D Missing Proofs from Section 4

Lemma 4.2 is highly non-trivial, and has therefore been assigned its own section (Section D.1) for readability.

Lemma 4.1. *There exists a polynomial time algorithm that computes a stable, max doctor-USW allocation. Moreover, the output allocation has the highest possible hospital-USW among all stable, max doctor-USW allocations.*

Proof. The ground set of both matroids is the set of all doctor hospital pairs $E = \{(d, h) : d \in D, h \in H\}$. In the first matroid, a set S is independent if the allocation X where $X_h = \{d : (d, h) \in S\}$ is non-redundant. The second matroid is a partition matroid where each part is of the form $\{(d, h) : h \in H\}$ for each $d \in D$. This matroid ensures that each doctor is allocated at most once. Each element (d, h) has weight $c_d(h)$.

It is easy to see that the max weighted independent set at the intersection of these two matroids corresponds to the non-redundant allocation with maximum doctor-USW. For this problem there exist efficient algorithms to compute the max weighted independent set subject to the independent set having size equal to some integer k [7]. The size of the independent set is equal to the hospital-USW of the output allocation. Therefore, by trying all m possible values of k , we can find the max doctor-USW allocation with the highest possible hospital-USW. $\square$

Theorem 4.3. *Algorithm 3 terminates in polynomial time and outputs an allocation X that maximizes doctor welfare and subject to that maximizes hospital Nash welfare.*

Proof. We first show that the algorithm terminates in polynomial time. To do this, we show that the **while** loop terminates in polynomial time. This is sufficient since we run the **while** loop for m different values of k .

We use a potential function argument to bound the number of **while** loop iterations. Every time we change the allocation, we strictly reduce $\sum_{i \in H} \left(|X_i| + \frac{i}{n^2} \right)^2$. Therefore, the while loop can only run $\mathcal{O}(m^2 n^2)$ times. We note that this exact potential function argument has been used before in Babaioff et al. [2] and Cousins et al. [9]

Next, we prove the correctness of our algorithm. Each allocation X^k computed by the algorithm maximizes doctor-USW subject to the constraint that total hospital-USW is k . Since we pick an allocation X^k that maximizes doctor-USW, the allocation X output by Algorithm 3 maximizes doctor-USW. Next, we show that X maximizes hospital Nash welfare. Let Y be an allocation that (a) maximizes doctor-USW, (b) hospital-NSW subject to (a), and (c) lexicographically dominates all other allocations that satisfy (a) and (b). Furthermore, assume that Y has a hospital social welfare of k .

Let us consider the allocation X^k that Algorithm 3 computes. Assume for contradiction that $X^k \neq Y$. Since $X^k \neq Y$, there is some hospital i such that $|X_i^k| < |Y_i|$. If there are multiple such hospitals, we break ties by choosing the i with lower $|X_i|$, and breaking further ties by choosing the lower i . Since both X^k and Y have a total hospital-USW of k , there is some hospital j such that $|Y_j| < |X_j^k|$; again, if there are multiple such hospitals, we break ties by choosing the j with lower $|Y_j|$, and breaking further ties by choosing the lower j .

We have four possible cases, each leading to a contradiction proving that $X^k = Y$.

Case 1: $|X_i^k| = |Y_j|$ and $i < j$. We apply Lemma 4.2 with the allocation X^k and hospital i to find a hospital ℓ and a k -MDW allocation Z such that the hospital i gets one more doctor, the hospital ℓ gets one less doctor and all other hospitals have the same bundle size. We have $|X_\ell^k| - 1 \geq |Y_\ell| \geq |Y_j| = |X_i^k|$. If any of these inequalities is strict, then $|X_\ell^k| \geq |X_i^k| + 2$; in this case, Algorithm 3 reduces the capacity of ℓ by 1 and increases the capacity of i by 1, and moves from the allocation X^k to Z , which contradicts X^k being the resulting allocation. If equality holds throughout, then $|X_\ell^k| = |X_i^k| + 1$ and $\ell > j > i$, which again contradicts the logic of Algorithm 3.

Case 2: $|X_i^k| < |Y_j|$. This case follows a similar argument to that of Case 1.

Case 3: $|X_i^k| = |Y_j|$ and $j < i$. In this case, we plug in Lemma 4.2 with allocation Y and hospital j to find a hospital ℓ and a k -MDW allocation Z such that j gets one more doctor, ℓ gets one less doctor and all other hospitals have the same bundle size. We have $|Y_\ell| - 1 \geq |X_\ell^k| \geq |X_i^k| = |Y_j|$. If any of these inequalities is strict, then $|Y_\ell| \geq |Y_j| + 2$ which means Z has a higher hospital Nash welfare than Y , a contradiction to Y being hospital-NSW optimal. If equality holds throughout, then $|Y_\ell| = |Y_j| + 1$ and $\ell > i > j$, which implies that Z is hospital-NSW optimal as well but lexicographically dominates Y , which again violates our assumption on Y .

Case 4: $|X_i^k| > |Y_j|$. The proof of this case follows a similar argument to that of Case 3.

Therefore, $X^k = Y$, and the Nash welfare of our output allocation X is at least that of Y . $\square$

Theorem 4.4. *The problem of deciding whether there exists a stable allocation where all hospitals are allocated at least one doctor is NP-complete when hospitals have capped binary additive valuations.*

Proof. We reduce from the NP-complete 2P2N-3SAT problem [6]. This class comprises of 3SAT instances with m clauses over variables $(x_1, \dots, x_n)$, such that each variable appears *exactly* two times in positive form and two times in negative form. An instance is a “yes” instance if and only if it has a valid truth assignment. Equation (4) is an example instance of 2P2N-3SAT with $n = 3$ variables and $m = 4$ clauses.

$$\overbrace{(x_1 \vee x_2 \vee \neg x_3)}^{C_1} \wedge \overbrace{(\neg x_1 \vee x_2 \vee x_3)}^{C_2} \wedge \overbrace{(\neg x_1 \vee \neg x_2 \vee \neg x_3)}^{C_3} \wedge \overbrace{(x_1 \vee \neg x_2 \vee x_3)}^{C_4} \quad (4)$$

Given a 2P2N-3SAT instance, our reduction is as follows: we have $m + 3n$ hospitals and $12n$ doctors. We create for each variable x_i , two copies of a doctor corresponding to the positive literal of the variable (x_i, x'_i) , and two copies of the doctor corresponding to the negative literal of the variable $(\neg x_i, \neg x'_i)$. We also create eight dummy doctors for each variable $(a_i^+, a_i^-, b_i^+, b_i^-, c_i^+, c_i^-, d_i^+, d_i^-)$.

We create one hospital h_j corresponding to each clause C_j . The hospital h_j has a capacity of 1 and values each literal in the corresponding clause at 1, all copies included. We also create two hospitals s_i^+, s_i^- for each variable x_i who we call the positive sink and the negative sink respectively. These two sink hospitals have a capacity of 4 each. The positive sink s_i^+ values the doctors $x_i, x'_i, a_i^+, b_i^+, c_i^+$ and d_i^+ at 1; similarly, the negative sink s_i^- values the doctors $\neg x_i, \neg x'_i, a_i^-, b_i^-, c_i^-$ and d_i^- at 1. We also create a priority hospital p_i for each variable x_i ; the priority hospital for variable x_i values the doctors $a_i^+, a_i^-, b_i^+, b_i^-, c_i^+, c_i^-$ at 1 and has a capacity of 3.

Coming to the preferences of the doctors, the four positive dummy doctors of each variable x_i ($a_i^+, b_i^+, c_i^+, d_i^+$) rank their priority hospital p_i highest, then their positive sink s_i^+ , and then all other hospitals. The four negative dummy doctors of each variable ($a_i^-, b_i^-, c_i^-, d_i^-$) rank their priority hospital p_i highest, then their *negative* sink s_i^- , and then all other hospitals.

The positive literals of each variable prefer their corresponding positive sink s_i^+ to any other hospital and the negative literals of each variable prefer their corresponding negative sink s_i^- to any other hospital.

Any ranking relation not mentioned in this construction can be filled up arbitrarily. This construction can be done in polynomial time, and its size is polynomial in the size of the original 2P2N-3SAT instance.

The crucial property maintained by the construction is that for any stable allocation, if we restrict ourselves to the allocations to the clause hospitals $h_1, \dots, h_m$, a variable can be allocated in positive form or negative form *but not both*. This is because to allocate a positive (negative) literal of a variable x_i to a clause hospital h_j , the positive (resp. negative) sink hospital of that variable must be filled up, i.e., the positive (resp. negative) sink must receive its capacity of four doctors to not create a blocking pair: otherwise, since the positive (resp. negative) sink hospital values the positive (resp. negative) literal doctor at 1, they form a blocking pair. To fill up either sink, the priority hospital p_i for that variable must be filled up; this is because all the doctors who are valued by the sink hospitals are also valued by the priority hospitals, whom these doctors strongly prefer. The total capacity of s_i^+, s_i^- and p_i is 11 (the positive and negative sinks have a capacity of 4 and the priority hospital has a capacity of 3); however, they together value 12 doctors at 1 — the four literal doctors $x_i, x'_i, \neg x_i, \neg x'_i$ and the eight dummy doctors — filling them up means there is only one doctor (either a positive or negative literal) corresponding to that variable that can be allocated to the clauses. Therefore, the allocation to the clauses cannot have two different kinds of literals for the same variable. This gives us the consistency needed to map stable allocations to SAT solutions.

The rest of the proof is straightforward: if there is a solution to the 2P2N-3SAT instance, then there is a straightforward allocation where all hospitals get a utility of at least 1. If a variable x_i is positive in the original solution, we fill up the priority hospital with a_i^-, b_i^-, c_i^- , the positive sink with a_i^+, b_i^+, c_i^+ and

d_i^+ , and then allocate the doctors corresponding to the positive literals to the clauses which have these literals. We can do this for all the variables. If there are multiple literals which satisfy a clause, we can pick one arbitrarily. Note that this allocation gives all the clauses a positive utility. The negative and positive sinks have a positive utility in any stable allocation because of the doctors d_i^+ and d_i^- who can only be allocated to these hospitals.

Assume there is no solution to the 2P2N-3SAT instance. In any stable allocation, from our above discussion, the set of hospitals corresponding to clauses must receive either the positive form of a literal or the negative form of a literal but not both. This choice can be thought of as an assignment to the variables. Since the 2P2N-3SAT instance is unsatisfiable, there is at least one clause that is not satisfied by this assignment; this clause corresponds to a hospital which receives a utility of 0 in the stable allocation. Since all assignments are unsatisfiable, all stable allocations give at least one hospital a utility of 0. $\square$

D.1 Proof of Lemma 4.2

The problem of maximizing doctor-USW can be modeled as a weighted matroid intersection problem over the ground set $E = \{(h, d) | d \in D, h \in H\}$ ³. Any assignment of doctors to hospitals can be equivalently defined as a subset of the ground set E . Specifically, each allocation X can be denoted by the set $\bigcup_{d \in D} (X(d), d)$, which is a subset of E . Similarly, any subset of E can be written as an allocation, although this allocation might allocate a doctor multiple times.

Given an allocation X represented as a subset of the ground set E , we construct the *matroid exchange graph* $\mathcal{G}(X)$. $\mathcal{G}(X)$ is a directed bipartite graph over the set of nodes E . The nodes of this graph are hospital-doctor pairs, and there are two types of edges. An edge exists from the node $(h, d) \in X$ to $(h', d') \in E \setminus X$ if swapping (h, d) with (h', d') in the allocation X does not violate the non-redundancy of the allocation. In other words, withholding d from h and assigning d' to h' instead results in a non-redundant allocation. An edge exists from $(h', d') \in E \setminus X$ to $(h, d) \in X$ if swapping the two in the allocation X does not result in any doctor being allocated to multiple hospitals. Any edge from $(h, d) \in X$ to $(h', d') \in E \setminus X$ has a weight of $c_d(h) - c_{d'}(h')$. All other edges have a weight of 0.

For any $S \subseteq X$ and $S' \subseteq E \setminus X$ such that $|S'| = |S|$, a matching M is a set of $|S|$ edges in $\mathcal{G}(X)$ from S to S' such that every node in $S \cup S'$ is connected to at least one edge in M . A back matching M' is a set of $|S|$ edges from S' to S such that every node in $S \cup S'$ is connected to at least one edge. We say the set (S, S') is a valid swap if $(X \setminus S) \cup S'$ is a valid allocation — it is non-redundant and each doctor is allocated at most once. We refer to the allocation $(X \setminus S) \cup S'$ as the allocation generating by *performing* the valid swap (S, S') on the allocation X . We slightly abuse notation if the matching and back matching form a cycle C : we say the cycle C is a valid swap if the nodes in the cycle form a valid swap. We similarly define the act of performing the valid swap C on an allocation X .

We utilize the following results on exchange graphs from Brezovec et al. [7].

Theorem D.1 (Brezovec et al. [7]). *Let X and Y be two k -MDW allocations*

- (a) *There exists a matching and a back matching in the exchange graph $\mathcal{G}(X)$ between $X \setminus Y$ and $Y \setminus X$.*
- (b) *There is no negative weight cycle in the exchange graph $\mathcal{G}(X)$.*
- (c) *Consider any sets $S \subseteq X$ and $S' \subseteq E \setminus X$ such that $|S| = |S'|$. Assume there is a matching and back-matching between these two sets in $\mathcal{G}(X)$. Then either (S, S') is a valid swap, or there is another matching and back matching between these two sets which are not identical to the first matching and back matching respectively.*

³An explanation for this statement can be found in the proof of Lemma 4.1 in Appendix D.

- (d) Let there be two matchings and back matchings between two sets $S \subseteq X$ and $S' \subseteq E \setminus X$ such that each matching and back matching corresponds to a single cycle. We can decompose these two cycles (say C and C') into a set of cycles $C_1, C_2, \dots, C_u$ such that each new cycle has length smaller than $2|S|$. Moreover, each edge appears the same number of times in $C_1, \dots, C_u$ as it appears in the two cycles C and C' .

All four statements above can be found in Brezovec et al. [7]: (a) is equivalent to Lemma 2, (b) is equivalent to Theorem 3, (c) is equivalent to Lemma 3, and (d) is equivalent to Lemma 4.

Consider any two k -MDW allocations X and Y . From Theorem D.1(a), a matching and back-matching must exist between $X \setminus Y$ and $Y \setminus X$. The matching and back matching form a set of disjoint cycles $\mathcal{C}$.

We say an edge from $(h, d) \in X$ to $(h', d') \in E \setminus X$ is a *cross edge* if $h \neq h'$. This essentially means that $\Delta_{h'}(X_{h'}, d') = 1$. Therefore, there must exist an edge from all $(h'', d'') \in X$ to (h', d') in $\mathcal{G}(X)$. We use this observation for our next operation, which we call *uncrossing*.

Lemma D.2 (Uncrossing Lemma). *For some allocation, let C be a cycle in the exchange graph $\mathcal{G}(X)$ over the set of nodes S . There exist a set of disjoint cycles $C_1, C_2, \dots, C_u$ in $\mathcal{G}(X)$ such that each node in S is contained in exactly one cycle and each cycle has at most one cross edge.*

Proof. Assume that the cycle C has more than one cross-edge. Suppose that C consists of the ordered set of edges $\{e_1, e_2, \dots, e_s\}$, where e_1 is a cross edge and the next cross edge after e_1 is at e_t . Let e_1 be an edge from (h_1, d_1) to (h_2, d_2) , and let e_t be an edge from (h_3, d_3) to (h_4, d_4) .

We replace e_1 with e'_1 which is an edge from (h_1, d_1) to (h_4, d_4) . We replace e_t with e'_t which is an edge from (h_3, d_3) to (h_2, d_2) . Both e'_1 and e'_t must exist in $\mathcal{G}(X)$ since e_1 and e_t are cross edges. We then output the cycles $C_1 = \{e_2, e_3, \dots, e_{t-1}, e'_t\}$ and $C_2 = \{e'_1, e_{t+1}, \dots, e_s\}$. C_1 has at most one cross edge; the only possible cross edge is e'_t . We then repeat this procedure with C_2 till we have a disjoint set of cycles, each with at most one cross-edge. $\square$

We are now ready to prove the lemma.

Lemma 4.2. *Let X and Y be two k -MDW allocations. Then, for any hospital $i \in H$ such that $|X_i| < |Y_i|$, there exists a hospital $j \in H$ such that $|X_j| > |Y_j|$ and a k -MDW allocation X' such that*

- (i) $|X'_i| = |X_i| + 1$,
- (ii) $|X'_j| = |X_j| - 1$, and
- (iii) $|X'_p| = |X_p|$ for all $p \neq i, j$.

Proof. From Theorem D.1(a), there is a matching and back matching between $X \setminus Y$ and $Y \setminus X$ in the exchange graph $\mathcal{G}(X)$. This matching and back matching can be assumed to be a set of cycles $\mathcal{C}$ with at most one cross edge in each cycle (Lemma D.2).

We associate each cycle with a winning and losing hospital. Suppose that the cycle C corresponds to a valid swap. A hospital is winning with respect to a cycle C if its utility strictly increases if the swap is executed. A losing hospital is one whose utility strictly decreases if this happens. Since each cycle has at most one cross edge, there is at most one winner and loser in a cycle. Even if C is not a valid swap, we define winners and losers by pretending it is a valid swap and looking at the size of each hospital's bundle before and after the swap.

The total weight of all the cycles in $\mathcal{C}$ is 0. This follows from the fact that both X and Y are k -MDW allocations, and the total weight of all the cycles in $\mathcal{C}$ is equal to $\sum_{(h,d) \in X \setminus Y} c_d(h) - \sum_{(h',d') \in Y \setminus X} c_{d'}(h')$. Since there is no negative weight cycle in the graph (Theorem D.1(b)), each cycle in $\mathcal{C}$ has a weight of 0. By the assumptions made in the lemma, $|Y_i| > |X_i|$; so there must be (at least) one cycle C in $\mathcal{C}$ where

hospital i is a winner. We show that either C is a valid swap, or we can find a smaller 0 weight cycle where i is the winner.

If C is a valid swap, then we are done. If C is not a valid swap, then from Theorem D.1(c), there is another matching and back matching between the nodes of C . We then apply the uncrossing operation to this new matching and back matching to create a new set of cycles $C_1, \dots, C_s$. Since C is a 0 weight cycle, each of $C_1, \dots, C_s$ must also be a 0 weight cycle (Theorem D.1(b)).

Case 1: $s > 1$. There must be a cycle C' in this new set of cycles such that i is the winner. Note that C' is smaller than C and C' has 0 weight.

Case 2: $s = 1$. We invoke Theorem D.1(d) to rewrite these two cycles C and C_1 as a set of cycles $C'_1, \dots, C'_u$ where each C'_w is smaller than C . We then apply the uncrossing lemma to each of these cycles to ensure that there is at most one cross edge in each of the cycles. Then we find the cycle C' in this new set of cycles where i is the winner. C' is smaller than C . C' also has 0 weight since both C and C_1 have 0 weight, and there are no negative weight cycles in $\mathcal{G}(X)$ (Theorem D.1(b)).

In summary, either the cycle we find is a valid swap and has 0 weight, Or we find a smaller cycle with 0 weight where i is the winner. We can repeat the above procedure with the smaller cycle until we find a cycle C^* which is a valid swap. We perform this valid swap using C^* on the allocation X to create a k -MDW allocation Z where i 's utility increases by 1, some hospital q 's utility decreases by 1 and all the other hospitals have the same utility.

If $|X_q| > |Y_q|$, we are done. Otherwise, we repeat this procedure with the allocation Z and q . This procedure is guaranteed to terminate, since with each new allocation we strictly increase $|X \cap Y|$; that is $|Z \cap Y| > |X \cap Y|$. $\square$