Provision-After-Wait with Common Preferences

HAU CHAN, Trinity University
JING CHEN, Stony Brook University
GOWTHAM SRINIVASAN, Stony Brook University

In this paper, we study the Provision-after-Wait problem in healthcare (Braverman, Chen, and Kannan, 2016). In this setting, patients seek a medical procedure which can be performed by different hospitals of different costs. Each patient has a value for each hospital and a budget-constrained government/planner pays for the expenses of the patients. Patients are free to choose hospitals, but the planner controls how much money each hospital gets paid and thus how many patients each hospital can serve (in one budget period, say one month or one year). Waiting times are used in order to balance the patients' demand, and the planner's goal is to find a stable assignment that maximizes the social welfare while keeping the expenses within the budget. It has been shown that the optimal stable assignment is NP-hard to compute, and beyond this, little is known about the complexity of the Provision-after-Wait problem.

We start by showing that this problem is in fact strongly NP-hard, thus does not have an FPTAS. We then focus on the *common preference* setting, where the patients have the same ranking over the hospitals. Even when the patients perceive the hospitals' values to them based on the same quality measurement —referred to as *proportional preferences*, which has been widely studied in resource allocation— the problem is still NP-hard. However, in a more general setting where the patients are ordered according to the differences of their values between consecutive hospitals, we construct an FPTAS for it. To develop our results, we characterize the structure of optimal stable assignments and their social welfare, and we consider a new combinatorial optimization problem which may be of independent interest, the *ordered Knapsack* problem.

Optimal stable assignments are deterministic and ex-post individually rational for patients. The downside is that waiting times are dead-loss to patients and may burn a lot of social welfare. If randomness is allowed, then the planner can use lotteries as a rationing tool: the hope is that they reduce the patients' waiting times, although they are interim individually rational instead of ex-post. Previous study has only considered lotteries for two hospitals. In our setting, for arbitrary number of hospitals, we characterize the structure of the optimal lottery scheme and conditions under which using lotteries generates better (expected) social welfare than using waiting times.

CCS Concepts: •**Theory of computation → Design and analysis of algorithms; Approximation algorithms analysis; Packing and covering problems; Algorithmic game theory and mechanism design;**

Additional Key Words and Phrases: budget, resource allocation, healthcare, waiting time, lottery

1. INTRODUCTION

In this paper we consider the Provision-after-Wait problem in healthcare [Braverman et al. 2016]. The problem studies the interaction among the patients, the hospitals, and a planner such as the government. Each patient seeks a non-urgent medical service, say X-ray or MRI, and has different values for different hospitals. Each hospital has a cost for serving one patient, which must be paid. The patients do not pay for the service and, instead, the planner pays for all of them (e.g., in the public sectors

---

A conference version of this paper appeared at the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'16) [Chan and Chen 2016]. We thank several anonymous reviewers for their helpful comments. The second author is partially supported by NSF CAREER Award No. 1553385.

DOI: http://dx.doi.org/10.1145/0000000.0000000

of healthcare). However, the planner has a budget for how much he can spend and might not be able to afford the costs incurred by all patients going to their most preferred hospitals. The planner needs to decide how to distribute his budget among the hospitals, and thus how many patients he can afford each hospital to serve (in one budget period, say a month or a year). Patients choose their favorite hospitals and, if a hospital is over-demanded, then a waiting-time is specified for it: the amount of time each patient has to wait before getting served there. Given the waiting times, a patient chooses the hospital that maximizes his utility, which is his value for the hospital minus the waiting time there.[1] De facto, the planner uses waiting times as a rationing tool and, when the patients pick their utility-maximizing hospitals, the total cost is within the budget. In addition, each patient should have a non-negative utility at his chosen hospital. We call such a solution a *stable* assignment (thus a stable assignment is automatically envy-free). As waiting times are dead-loss to the patients, the *social welfare* is the total utility of the patients rather than total value. Among all stable assignments, we are interested in finding the ones that maximize the social welfare. The same social welfare was considered by [Hartline and Roughgarden 2008] in a different model. Even when each patient i has value 0 for all hospitals but hospital i, the problem of computing the optimal stable assignment is NP-hard [Braverman et al. 2016]. The authors also provide an algorithm that may cause a small multiplicative deficit to the planner's budget. Beyond this, little is known about the computational complexity of this problem.

In the technical part of this paper, we start with a simple theorem, Theorem 4.1, showing that the optimal stable assignment is actually strongly NP-hard to compute. As an immediate consequence, there is no Fully Polynomial-Time Approximation Scheme (FPTAS) for the problem. Thus, it is natural to look for interesting and more structured settings where the problem can be solved efficiently.

In this paper, we consider the *common preference* setting which has been widely studied in the literature of resource allocation: that is, using the language of the Provision-after-Wait problem, the patients have the same ranking over the hospitals. In particular, this includes *proportional preferences* as special cases, where the patients perceive the hospitals' values to them based on the same quality measurement. More precisely, each patient has a value for receiving the medical service, each hospital has a quality factor that is publicly known, and a patient's value for a hospital is the product of the two. There have been many studies on proportional preferences in other resource allocation problems. For instance, in auctions for advertisement slots [Athey and Ellison 2009; Varian 2007], the quality of a slot is its view-through or click-through rate. In [Devanur et al. 2013; Ha and Hartline 2013; Hartline and Yan 2011], the quality of a resource represents the probability of obtaining the resource. [Alaei et al. 2014] considers another economic setting with proportional values. As for hospitals, the *star rating* is a particular example of quality factors, and patients can obtain this and other quality measurements from, say, US News Best Hospitals 2015-16[2] or Centers for Medicare and Medicaid Services of the US government[3].

In fact, the class of patients' preferences we study is much larger than proportional preferences. Formally defined in Section 3, we consider preferences where the patients are ordered according to the differences of their values between consecutive hospitals, and refer to them as *d-ordered* preferences (with "d" for "difference"). For example, it may be the case that patient 1's value difference for the first two hospitals is larger

[1]That is, a patient's value for a hospital is measured as "willingness to wait". This is similar to auctions, where a buyer's value is measured as "willingness to pay".
[2]http://health.usnews.com/best-hospitals.
[3]http://www.cms.gov/.

than that of patient 2, which is larger than that of patient 3, etc; and this is true for any two consecutive hospitals. It will become clear from the definition that proportional preferences are special cases of this class.

Computing Optimal Stable Assignments. In Section 5, we characterize the structure of optimal stable assignments and the corresponding social welfare. The characterization is a key tool for most of our results in Sections 5 and 6. Based on it, we first show that the problem is still NP-hard even with proportional preferences (Theorem 5.10), and then provide an FPTAS for d-ordered preferences (Theorem 5.14). More specifically, letting n and m respectively be the number of patients and the number of hospitals, the FPTAS runs in time $O((n+m)n^3m/\epsilon)$. Theorems 5.10 and 5.14 together give us a good understanding about the Provision-after-Wait problem with d-ordered preferences.

To construct the desired FPTAS, we introduce another problem, *ordered Knapsack*. Roughly speaking, this is a bounded Knapsack problem[4] where the items' values depend on the order under which the items are packed into the knapsack. We construct an FPTAS for this problem and use it to approximate the optimal stable assignment. We believe the ordered Knapsack problem itself is also worth further study.

Optimal Lottery Schemes. The optimal stable assignment is deterministic and ex-post individually rational for the patients. However, waiting times are dead-loss to patients and may burn a lot of social welfare. Another widely used rationing tool in resource allocation such as school choice is lotteries [Cullen et al. 2006; Lavy 2010; Ashlagi and Shi 2014; Deming et al. 2014], which allows the planner to randomly allocate resources to people, rather than giving them free choices. The hope is that lotteries can reduce waiting times and improve the (expected) social welfare, although they are only required to be interim individually rational instead of ex-post. Interestingly, in the real world, lotteries are widely used in school choice and waiting times are considered undesirable there; while in healthcare waiting times are widely used and, to our best knowledge, so far there is no healthcare system that relies on lotteries. This motivated the authors of [Braverman et al. 2016] to study the structure of optimal lottery schemes in healthcare, which combines waiting time and randominess, although they only considered the case of two hospitals.

Roughly speaking, a *lottery* is a distribution over hospitals together with a waiting time, and a *lottery scheme* is a set of lotteries for the patients to choose from. A patient choosing a particular lottery gets a sample from the distribution and goes to the specified hospital after the specified waiting time. Each patient chooses one lottery to maximize his expected utility. The social welfare and the budget constraint for a lottery scheme are also measured "in expectation."

It is easy to see that stable assignments are special cases of lottery schemes: there are m "lotteries" and lottery i assigns the patient to hospital i with probability 1 after the specified waiting time there. Another natural sub-class of lottery schemes are *randomized assignments*, which consists of a single lottery with waiting time 0. Accordingly, in a randomized assignment the patients do not have any choice and the planner just randomly assigns them to hospitals according to the corresponding distribution. We formally define these notions in Section 6. For now, it is worth pointing out that stable assignments and randomized assignments are extreme cases of lottery schemes: the former utilizes waiting times but not randomness, the latter utilizes randomness but not waiting times, while a general lottery scheme utilizes both.

[4]Different from the Knapsack problem where each item has a single copy, the bounded Knapsack problem allows each item to have multiple copies; see, e.g., [Martello and Toth 1990].

In principle, randomized assignments generate less social welfare than the optimal lottery scheme. However, they have the advantage of being efficiently computable, as we will see in Section 6. Thus, we are interested in understanding the relationship among optimal stable assignments, optimal randomized assignments, and optimal lottery schemes in general. For arbitrary number of hospitals and d-ordered preferences, we identify conditions under which optimal randomized assignments generate more (expected) social welfare than optimal stable assignments (Theorem 6.3). Surprisingly, for proportional preferences, the same condition implies that the optimal randomized assignment is actually optimal among all lottery schemes (Theorem 6.7). Our results significantly generalize that of [Braverman et al. 2016].

The optimal randomized assignment has many good properties, which we briefly mention below and discuss more carefully in Section 6.3. First, besides being efficiently computable, the randomized assignment can be implemented so that the budget constraint is satisfied with probability 1. Indeed, lotteries in general have the problem that all guarantees are "in expectation" and, in the worst case, the social welfare may be poor and the budget constraint may be violated. Accordingly, an important topic in resource allocation is to find lotteries that can be implemented with ex-post performance guarantees [Budish et al. 2013]. It remains an interesting open problem to design such lottery schemes in our setting for the budget constraint and the social welfare simultaneously.

Second and more interestingly, the conditions in Theorems 6.3 and 6.7 have a very natural interpretation from a different aspect. Indeed, we consider the patients as a continuous population over say, $[0, 1]$, and consider the random variable for patients' values for the hospitals induced by the uniform distribution over the patient population. Then, the conditions in Theorems 6.3 and 6.7 are exactly that the corresponding distributions of the value differences between consecutive hospitals have *monotone hazard rates* (MHR). This brings up an interesting connection between our result and lottery pricing schemes with a single buyer and multiple items [Chawla et al. 2007], where optimal pricing schemes are studied when the distributions of the buyer's values have MHR.

2. ADDITIONAL RELATED WORKS

The role of waiting time in healthcare has been extensively studied in the literature. Many works use it as a rationing tool to balance demand and supply for non-urgent medical services; and many others consider its welfare-burning effect and try to reduce waiting in healthcare. In particular, [Gravelle and Siciliani 2008b] uses both waiting time and service quality as instruments to control the demand in health insurance, while the instruments used in [Felder 2008] are waiting time and copayments. Their models and objectives are quite different from ours though. [Gravelle and Siciliani 2008a, 2009] use waiting time prioritisation/discrimination to improve social welfare. In particular, they consider setting different waiting times for different groups of patients requiring the same treatment, even when the patients are in the same waiting list. Different from their work, in our model the patients going to the same hospital all face the same waiting time. On the empirical front, [Dawson et al. 2007] provides experimental studies on reducing waiting times by offering patients who face a long wait at their desired hospital the choice of an alternative hospital with a guaranteed shorter wait. Moreover, [Siciliani and Hurst 2005] gives a thorough comparative analysis on waiting times of existing elective-surgery policies in OECD countries. Finally, it is worth pointing out that existing studies usually focus on a single healthcare system: in particular, the public sector. However, having waiting times for a service in one system do not necessarily mean hospitals or doctors are idle: in reality, they serve patients from multiple healthcare systems.

None of the above mentioned studies considers the budget constraint of the planner and, to our best knowledge, the effect of the planner's budget on waiting times was considered for the first time in [Braverman et al. 2016]. To help understanding the emergence of waiting times under the existence of budget, the authors consider the dynamic between patients and hospitals. Given how the planner's budget is distributed among the hospitals, when the patients arrive continuously along time, the dynamic starts with waiting time 0 everywhere. It lets patients choose hospitals freely and adjusts waiting times according to the demand. The authors show that when the patients are generic, the dynamic always converges to the optimal stable assignment and, after the convergence, all patients going to the same hospital face the same waiting time. Accordingly, the optimal stable assignment can be considered as the stable state of the dynamic, rather than being enforced exogenously by the planner.

Since the authors of [Braverman et al. 2016] allow arbitrary values of the patients for the hospitals, the NP-hardness for computing the optimal stable assignment there is much easier than in our setting. When there is a small number of hospitals, they provide an algorithm using budget $(1+\epsilon)B$ to generate a stable assignment whose social welfare is at least the optimal social welfare under budget B, but the technique is different from ours. Moreover, the authors study optimal lottery schemes only for two hospitals. In this paper, besides strengthening the hardness result for the general Provision-after-Wait problem, we focus on patients with common preferences and greatly improve the understanding on both stable assignments and lottery schemes.

Notice that in our model, after the planner has decided how to distribute the budget among the hospitals, the allocation of affordable slots at different hospitals to patients becomes a unit-demand auction, with waiting times as prices. However, in unit-demand envy-free pricing schemes the goal is usually to maximize revenue instead of social welfare. In [Devanur et al. 2013] and [Hartline and Yan 2011], the authors study pricing problems where the buyers have proportional values and characterize the optimal envy-free solutions for maximizing revenue and the total value of the players. While most works on pricing schemes study deterministic optimal item-pricing [Briest 2008; Chawla et al. 2007; Chen et al. 2014; Guruswami et al. 2005; Hartline and Yan 2011], a few consider lotteries [Briest et al. 2010; Chawla et al. 2010; Manelli and Vincent 2006; Thanassoulis 2004] and show that they can generate more revenue than deterministic item-pricing in various cases. However, the structures of optimal lottery pricing schemes are far from being well understood.

Finally, as pointed out by [Braverman et al. 2016], part of the difficulties using waiting times as a rationing tool comes from the fact that the planner's budget and the patients' waiting times are two unexchangeable "currencies", as patients do not care about the money paid to the hospitals. Thus, although the problem of assigning patients to hospitals looks like matching in labor markets between firms and workers, the approaches are very different. For example, in a classic study about matching in labor markets by [Kelso and Crawford 1982], the constraints include both the gross product and the salary of workers, but the former is measured in the same unit as the latter, thus there is only one "currency".

3. THE MODEL

In this section, we first review the Provision-after-Wait problem introduced by [Braverman et al. 2016] and then define the classes of patients' preferences considered in this paper. More specifically, there are n *patients*, indexed by $[n] = \{1, 2, ..., n\}$, and m *hospitals*, indexed by $[m] = \{1, 2, ..., m\}$. Each patient wants a single medical service, which can be provided by any one of the hospitals. For each hospital $j \in [m]$, there is a *cost*

$c_j \geq 0$ for serving one patient.[5] Each patient $i \in [n]$ has a value $v_{ij} \geq 0$ for receiving the service in hospital $j \in [m]$. As mentioned before, the patients do not pay for the service. Instead, the *planner* such as the government pays for everybody's service through some funding program (e.g., the Patient Protection and Affordable Care Act in United States—that is, Obamacare). The planner has a *budget* $B \geq 0$ that limits the total amount that he can spend, such that $n \min_{j\in[m]} c_j \leq B < n \max_{j\in[m]} c_j$. That is, the planner cannot afford to have all patients served at the most expensive hospital, but he can at least afford to serve all patients at the cheapest hospital. The planner thus decides a *quota* $\lambda_j \in \{0, 1, \ldots, n\}$ for each hospital j, indicating how much he can spend there, and thus how many patients this hospital can serve. A quota vector $\lambda = (\lambda_1, \ldots, \lambda_m)$ is *feasible* if $\sum_{j\in[m]} \lambda_j \geq n$ and $\sum_{j\in[m]} \lambda_j c_j \leq B$. Given the quotas, waiting times are used as a rationing tool to balance demand and supply. In particular, the problem becomes envy-free pricing, where each hospital j has λ_j "copies" for sale and has a *waiting time* $w_j \geq 0$ as its "price". For each patient i, her utility for going to hospital j is $v_{ij} - w_j$.

Given a quota vector λ, a solution to the Provision-after-Wait problem is called an *assignment*, which consists of a *waiting vector* $w = (w_1, \ldots, w_m)$ and an *assignment function* $a : [n] \to [m]$. For each patient i, $a(i)$ is the hospital that i goes to; and for each hospital j, $|a^{-1}(j)| \leq \lambda_j$. Given an assignment $A = (a, w)$, the *social welfare* of A is $SW(A) = \sum_{i\in[n]} v_{ia(i)} - w_{a(i)}$, the total *cost* of A is $C(A) = \sum_{i\in[n]} c_{a(i)}$, and A is *budget-feasible* if $C(A) \leq B$. An assignment A is *stable* if for any patient i, (1) $v_{ia(i)} - w_{a(i)} \geq 0$ and (2) for any hospital j, $v_{ia(i)} - w_{a(i)} \geq v_{ij} - w_j$. Note that we could have started by allowing different patients to have different waiting times at the same hospital, but envy-freeness automatically implies patients going to the same hospital have the same waiting time anyway.

The planner's goal is to find a feasible λ and a corresponding stable assignment A to maximize the social welfare. It is worth pointing out that in the real world the patients arrive along time and the waiting time at each hospital may change dynamically according to the demand. However, it is without loss of generality to consider a one-shot game with n patients and a fixed waiting time for each hospital. Indeed, as shown by [Braverman et al. 2016], given any quota vector λ, when there are n continuous patient populations arriving along time and when the patients' values are generic, the dynamic that starts with waiting time 0 at every hospital and adjusts waiting times according to the patients' demand always converges to the optimal stable assignment under λ. That is, waiting times derive endogenously from the budget constraint and the dynamic between patients and hospitals, and the static solution we are interested in here can be considered as the stable state of the dynamic. Note that before the dynamic converges, different patients arriving at different time points may face different waiting times at the same hospital, but at the stable state all patients going to the same hospital face the same waiting time. That is why in our solution we only specify one waiting time for each hospital.[6]

Moreover, given any stable and budget-feasible assignment $A = (a, w)$, the planner can always define λ to be $\lambda_j = |a^{-1}(j)|$ for each hospital j and λ is feasible. Therefore finding the optimal stable and budget-feasible assignment is the key problem and, in the discussion below, we will not explicitly specify λ and will focus on the assignment.

[5]That is, one hospital provides the same service to all patients, while different hospitals may provide different services. For example, one "hospital" may actually represent X-ray services and another one may represent MRI services.

[6]In particular, our study is different from waiting time discrimination [Gravelle and Siciliani 2008a, 2009], which sets different waiting times for different patients at the same hospital on purpose, so as to improve social welfare.

Definition 3.1. A stable assignment A is *optimal* if

$$A \in \underset{A' \text{ is stable and budget-feasible}}{\operatorname{argmax}} SW(A').$$

It is also useful to consider optimal stable assignments with respect to a particular assignment function, as follows.

Definition 3.2. For any assignment function a, a stable assignment $A = (a, w)$ is *optimal with respect to* a if $A \in \operatorname{argmax}_{A'=(a,w') \text{ and } A' \text{ is stable}} SW(A')$.

Note that budget-feasibility is not required in Definition 3.2: the cost of all assignments $A' = (a, w')$ is solely decided by a, thus either all of them are budget-feasible or none is. Below we introduce two classes of patients' preferences considered in this paper.

Proportional Preferences. As mentioned in the introduction, after showing that the general Provision-after-Wait problem is strongly NP-hard, we consider *common preferences* where the patients have the same ranking for the hospitals: without loss of generality, $v_{i1} \geq v_{i2} \geq \cdots \geq v_{im}$ for each patient i. A widely studied class of such preferences are *proportional preferences*, as follows. For each hospital j there is a quality factor $q_j \geq 0$ that is publicly known, and without loss of generality $q_1 \geq q_2 \geq \cdots \geq q_m$. For each patient i, there is a value $v_i \geq 0$ specifying i's value for being served, and i's value for each hospital j is $v_{ij} = v_i q_j$.

Note that $v_{i1} \geq v_{i2} \geq ... \geq v_{im}$ for each patient i, thus the patients have common preferences. By renaming the patients, $v_1 \geq v_2 \geq \cdots \geq v_n$ without loss of generality. Accordingly, we also have $v_{1j} \geq v_{2j} \geq \cdots \geq v_{nj}$ for each hospital j, and the matrix $v = (v_{ij})_{i \in [n], j \in [m]}$ has all rows and columns to be non-increasing. For clarity, when considering proportional preferences, we explicitly represent the value v_{ij} as $v_i q_j$.

Preferences That Are Ordered By Differences. A sub-class of common preferences that is much larger than proportional preferences are where the patients are ordered by differences, referenced to as *d-ordered preferences*. For each patient $i \in [n]$ and hospital $j \in \{2, ..., m\}$, letting $v^d_{ij} \triangleq v_{i(j-1)} - v_{ij} \geq 0$ be the difference of i's values for hospitals $j-1$ and j, we have the following.

Definition 3.3. The patients are *ordered by differences* (or *d-ordered* for short) if for each $j \in \{2, ..., m\}$, we have $v^d_{1j} \geq v^d_{2j} \geq \cdots \geq v^d_{nj}$.

That is, patient 1 is the most sensitive to the change from any hospital to the next best one, and then patient 2, etc. It is straightforward to verify that patients with proportional preferences are d-ordered, as $v^d_{ij} = (q_{j-1} - q_j)v_i$ for each patient i and hospital $j \geq 2$. However, in general d-ordered preferences $v = (v_{ij})_{i \in [n], j \in [m]}$ may not satisfy the additional property that $v_{1j} \geq v_{2j} \geq \cdots \geq v_{nj}$ for each hospital j.

Remark 3.4. For arbitrary d-ordered preferences $v = (v_{ij})_{i \in [n], j \in [m]}$, v may not satisfy the additional ordering property that $v_{1j} \geq v_{2j} \geq \cdots \geq v_{nj}$ for each hospital j as in proportional preferences. However, note that $v_{ij} = v^d_{i(j+1)} + \cdots + v^d_{im} + v_{im}$ for any patient i and hospital j and, if we define $v'_{ij} = v_{ij} - v_{im}$, then the resulting preferences $v' = (v'_{ij})_{i \in [n], j \in [m]}$ keep the original differences while satisfies the additional ordering property. Moreover, $v'_{im} = 0$ for any patient i. After our characterization in Section 5.1 for optimal stable assignments, it will become clear (see Section 5.3) that an optimal stable assignment for v is also an optimal stable assignment for v', and vice versa. Thus for computing optimal stable assignments it will be sufficient to consider d-ordered preferences of the type of v'. But we do not need this condition to establish the characterization.

4. THE COMPLEXITY OF THE GENERAL PROVISION-AFTER-WAIT PROBLEM

The following theorem holds from a simple reduction from Vertex Cover and motivates us to consider the Provision-after-Wait problem with common preferences.

THEOREM 4.1. *It is strongly NP-hard to compute an optimal stable assignment for the general Provision-after-Wait problem.*

PROOF. Consider the decision version of the general Provision-after-Wait problem:

$$\begin{aligned} DPaW \;=\; \{(c = (c_j)_{j\in[m]}, v = (v_{ij})_{i\in[n],j\in[m]}, B, T) :\ & \text{there exists a stable} \\ & \text{budget-feasible assignment } A \text{ such that } SW(A) \geq T\}. \end{aligned}$$

The Vertex Cover problem, which is well known to be strongly NP-hard, is defined as follows:

$$\begin{aligned} VC \;=\; \{(G = (V, E), k) :\ & \text{there exists } V' \subseteq V \text{ with } |V'| = k \\ & \text{such that, for each edge } \{u, v\} \in E, \{u, v\} \cap V' \neq \emptyset\}, \end{aligned}$$

where G is an undirected simple graph with vertex set V and edge set E. Given an instance (G, k) of VC with t vertices and e edges, and letting $V = \{1, \ldots, t\}$ and $E = \{\{u_1, v_1\}, ..., \{u_e, v_e\}\}$, we construct an instance (c, v, B, T) of $DPaW$ as follows.

— There are $t+1$ hospitals and $e+t$ patients. Each hospital $j \in [t]$ corresponds to a vertex $j \in V$ and hospital $t + 1$ corresponds to a dummy hospital. Each *edge-type* patient $i \in [e]$ corresponds to an edge $\{u_i, v_i\} \in E$ and each *vertex-type* patient $i \in [e + t] \setminus [e]$ corresponds to a vertex $i - e \in V$.
— For each hospital $j \in [t]$, $c_j = 1$; and for hospital $t + 1$, $c_{t+1} = 0$.
— For each edge-type patient $i \in [e]$, $v_{iu_i} = v_{iv_i} = t^2$ and $v_{ij} = 0$ for any other hospital $j \in [t + 1] \setminus \{u_i, v_i\}$. That is, i only wants hospitals corresponding to the vertices of its edge.
— For each vertex-type patient $i \in [e + t] \setminus [e]$, $v_{i(i-e)} = 1$ and $v_{ij} = 0$ for any other hospital $j \in [t+1] \setminus \{i-e\}$. That is, i only wants the hospital corresponding to its own vertex.
— Finally, $B = e + k$ and $T = et^2 + k$.

It is easy to see that the reduction takes polynomial time and produces an instance of $DPaW$ where all the parameters are polynomial in the size of (G, k). We have the following two lemmas.

LEMMA 4.2. $(G, k) \in VC \Rightarrow (c, v, B, T) \in DPaW$.

PROOF. Letting V' be a vertex cover of G with $|V'| = k$, we construct an assignment $A = (a, w)$ as follows.

— $w_j = 0$ for each hospital $j \in V' \cup \{t + 1\}$ and $w_j = t^2$ for each hospital $j \in V \setminus V'$.
— For each vertex-type patient $i \in [e + t] \setminus [e]$, if $i - e \in V'$ then $a(i) = i - e$, otherwise $a(i) = t + 1$.
— For each edge-type patient $i \in [e]$, if $u_i \in V'$ then $a(i) = u_i$; otherwise $a(i) = v_i$ (and $v_i \in V'$).

It is easy to see that the construction of A takes polynomial time. To see why A is budget-feasible, notice that there are exactly k vertex-type patients and e edge-type patients served in hospitals $\{1, \ldots, t\}$, with cost 1 each, thus the total cost is $e + k = B$.

Now we show that A is stable. For each edge-type patient $i \in [e]$, by the definition of a vertex cover we have $a(i) \in V'$, thus $w_{a(i)} = 0$ and $v_{ia(i)} - w_{a(i)} = t^2$, which is the maximum utility i can get from any hospital. For each vertex-type patient $i \in [e+t]\setminus[e]$ such that $i - e \in V'$, we have $v_{ia(i)} - w_{a(i)} = v_{i(i-e)} - w_{i-e} = 1 - 0 = 1$, which is again the maximum utility i can get from any hospital. Moreover, for each vertex-type patient i such that $i - e \notin V'$, we have $w_{i-e} = t^2$, $v_{i(i-e)} - w_{i-e} = 1 - t^2 < 0$, and $v_{ia(i)} - w_{a(i)} = v_{i(t+1)} - w_{t+1} = 0 \geq v_{ij} - w_j$ for each $j \in [t+1]$. Accordingly, the assignment A is stable.

Finally, the social welfare of A is

$$\begin{aligned} SW(A) &= \sum_{i\in[e+t]} v_{ia(i)} - w_{a(i)} = \sum_{i\in[e]} v_{ia(i)} - w_{a(i)} + \sum_{i\in[e+t]\setminus[e]} v_{ia(i)} - w_{a(i)} \\ &= et^2 + \sum_{i\in[e+t]\setminus[e],\ i-e\in V'} 1 = et^2 + |V'| = et^2 + k = T. \end{aligned}$$

In sum, $(c, v, B, T) \in DPaW$ as desired. $\square$

LEMMA 4.3. $(c, v, B, T) \in DPaW \Rightarrow (G, k) \in VC$.

PROOF. Letting $A = (a, w)$ be the optimal stable budget-feasible assignment, we have $SW(A) \geq T = et^2 + k$. Letting V' be the set of vertices whose corresponding hospitals have waiting time 0 in A, we show that V' is a vertex cover of G with size k.

First of all, since A is budget-feasible and each hospital in $[t]$ has cost 1, there can be at most $e + k$ patients served at these hospitals. Second, notice that each edge-type patient values a hospital for at most t^2 and each vertex-type patient values a hospital for at most 1. In order to achieve $SW(A) \geq et^2 + k$, it must be the case that all e edge-type patients and exactly k vertex-type patients are served by hospitals in $[t]$: if there are $x < e$ edge-type patients and y vertex-type patients served by hospitals in $[t]$, then the social welfare can be at most $xt^2 + y \leq xt^2 + t \leq (e-1)t^2 + t^2 = et^2 < et^2 + k$, a contradiction. Moreover, each of the k vertex-type patient is served at a hospital he values for 1, each edge-type patient is served at a hospital which he values for t^2, and all such hospitals have waiting time 0 in A. Accordingly, these hospitals are all in V'. By construction, each edge has a vertex in V' and V' is a vertex cover of G. Since each vertex-type patient corresponds to a different vertex, $|V'| \geq k$.

Finally, it is easy to see that $|V'|$ cannot be larger than k: otherwise, since A is stable, all the $|V'|$ corresponding vertex-type patients are served by hospitals in V' and the total cost is $e + |V'| > e + k = B$, a contradiction. Therefore we have $|V'| = k$ and $(G, k) \in VC$ as desired. $\square$

Theorem 4.1 follows directly from Lemmas 4.2 and 4.3.

5. OPTIMAL STABLE ASSIGNMENTS

From now on, we solely consider d-ordered preferences and may not explicitly mention this fact anymore.

5.1. The Structure of Optimal Stable Assignments

We start by characterizing the structure of optimal stable assignments and their social welfare. This characterization gives us the key tools for analyzing optimal stable assignments and is used in almost all of our main results. The following claims hold trivially and clarify several properties of the problem in our setting.

CLAIM 5.1. *When the patients are d-ordered, for any two patients $i < i'$ and two hospitals $j < j'$, $v_{ij} - v_{ij'} \geq v_{i'j} - v_{i'j'}$.*

PROOF. By Definition 3.3, $v_{ij} - v_{ij'} = \sum_{l=j+1}^{j'} v^d_{il} \geq \sum_{l=j+1}^{j'} v^d_{i'l} = v_{i'j} - v_{i'j'}$. □

CLAIM 5.2. *When the patients are d-ordered, for any stable assignment $A = (a, w)$ and any two hospitals $j < j'$ such that $a^{-1}(j') \neq \emptyset$, we have $w_j \geq w_{j'}$.*

PROOF. Let $i \in [n]$ be such that $a(i) = j'$. By definition, $v_{ij'} - w_{j'} \geq v_{ij} - w_j$, that is, $w_j - w_{j'} \geq v_{ij} - v_{ij'}$. Since $j < j'$ and the patients have common preferences, $v_{ij} \geq v_{ij'}$. Thus $w_j - w_{j'} \geq 0$ and Claim 5.2 holds. □

Below is an important definition.

Definition 5.3. An assignment function a is *ordered* if $a(1) \leq a(2) \leq \cdots \leq a(n)$.

The lemma below shows that it is sufficient to consider stable assignments $A = (a, w)$ where a is ordered.

LEMMA 5.4. *When the patients are d-ordered, given any stable assignment $A = (a, w)$, in polynomial time it can be modified so that: a is ordered, A is still stable, and the total cost and the utility of each patient remain the same.*

PROOF. Assume $a(i) > a(i')$ for some patients $i < i'$. Since the patients are d-ordered, $v^d_{ij} \geq v^d_{i'j}$ for all $j = 2, ..., m$. By the definition of stable assignments, we have

$$v_{ia(i)} - w_{a(i)} \geq v_{ia(i')} - w_{a(i')} \quad \text{and} \quad v_{i'a(i')} - w_{a(i')} \geq v_{i'a(i)} - w_{a(i)}. \tag{1}$$

Adding the two inequalities side by side and rearranging the terms, we have

$$\begin{aligned} & v_{ia(i)} - w_{a(i)} + v_{i'a(i')} - w_{a(i')} \geq v_{ia(i')} - w_{a(i')} + v_{i'a(i)} - w_{a(i)} \\ \iff \quad & v_{ia(i)} + v_{i'a(i')} \geq v_{ia(i')} + v_{i'a(i)} \\ \iff \quad & v_{i'a(i')} - v_{i'a(i)} \geq v_{ia(i')} - v_{ia(i)}. \end{aligned}$$

From Claim 5.1, since $i < i'$ and $a(i') < a(i)$, we have $v_{ia(i')} - v_{ia(i)} \geq v_{i'a(i')} - v_{i'a(i)}$. Together with the inequality above, we have $v_{ia(i')} - v_{ia(i)} = v_{i'a(i')} - v_{i'a(i)}$. Using (1), we have

$$w_{a(i')} - w_{a(i)} \geq v_{ia(i')} - v_{ia(i)} = v_{i'a(i')} - v_{i'a(i)} \geq w_{a(i')} - w_{a(i)}.$$

Since the leftmost equation is the same as the rightmost, both inequalities must be equalities. Accordingly,

$$v_{ia(i)} - w_{a(i)} = v_{ia(i')} - w_{a(i')} \quad \text{and} \quad v_{i'a(i')} - w_{a(i')} = v_{i'a(i)} - w_{a(i)}.$$

That is, patient i has the same utility at $a(i')$ and $a(i)$, so does patient i'. Thus, we can switch their assigned hospitals while keeping the stability of the assignment, obtaining the same total cost and patients' utilities as before, and having $a(i) \leq a(i')$. This immediately implies that $n-1$ switches suffice to make a ordered and Lemma 5.4 holds. □

By Lemma 5.4, we can focus on stable assignments $A = (a, w)$ where a is ordered. Note that for any such A, by Claim 5.2 we have

$$w_{a(1)} \geq w_{a(2)} \geq \cdots \geq w_{a(n)}. \tag{2}$$

Arbitrarily fixing an ordered assignment function a, we consider all the stable assignments with respect to a. Below is another important definition.

Definition 5.5. For any ordered assignment function a, the *tight assignment at a* is defined to be the assignment $A = (a, w)$ where $w_{a(n)} = 0$, $w_{a(i)} = v_{(i+1)a(i)} - v_{(i+1)a(i+1)} + w_{a(i+1)}$ for any $i < n$, and $w_j = +\infty$ for any $j \notin a([n])$.

If a stable assignment $A' = (a, w')$ is such that $w'(a(i)) = w(a(i))$ for each patient i, then we say that A' is *tight at* a.

In the definition above, w_j with $j \notin a([n])$ can be any sufficiently large number such that no patient wants to be served at hospital j. For example, it suffices to take $w_j = \max_{i\in[n]} v_{i1}$. In this sense, there are many tight assignments at a, but they can all be considered as the same. Since a is ordered, $a(i) \leq a(i+1)$ for each $i < n$. Since the patients have common preferences, $v_{(i+1)a(i)} \geq v_{(i+1)a(i+1)}$ for each $i < n$. Thus the $w_{a(i)}$'s in Definition 5.5 are all non-negative and A is well defined. Moreover, note that for any $i < n$, $v_{(i+1)a(i+1)} - w_{a(i+1)} = v_{(i+1)a(i)} - w_{a(i)}$: that is, patient $i+1$ is indifferent between hospitals $a(i+1)$ and $a(i)$. The following lemmas show that the tight assignment at a is not only stable but also optimal with respect to a. Moreover, it is essentially the only optimal stable assignment with respect to a.

LEMMA 5.6. *For any ordered assignment function a, the tight assignment at a is stable.*

PROOF. Letting $A = (a, w)$ be the tight assignment at a, we first compare i's utility at $a(i)$ with his utility at $a(i')$ for all $i' \neq i$.

For any $i' < i$, by Definition 5.5 we have

$$v_{(i'+1)a(i')} - w_{a(i')} = v_{(i'+1)a(i'+1)} - w_{a(i'+1)}. \tag{3}$$

Since $i'+1 \leq i$ and $a(i') \leq a(i'+1)$, by Claim 5.1 we have $v_{(i'+1)a(i')} - v_{(i'+1)a(i'+1)} \geq v_{ia(i')} - v_{ia(i'+1)}$, that is,

$$v_{(i'+1)a(i')} - v_{ia(i')} \geq v_{(i'+1)a(i'+1)} - v_{ia(i'+1)}. \tag{4}$$

Subtracting corresponding sides of (4) from those of (3), we have

$$v_{ia(i')} - w_{a(i')} \leq v_{ia(i'+1)} - w_{a(i'+1)}.$$

Since this holds for all $i' < i$, we have

$$v_{ia(1)} - w_{a(1)} \leq v_{ia(2)} - w_{a(2)} \leq \cdots \leq v_{ia(i-1)} - w_{a(i-1)} = v_{ia(i)} - w_{a(i)},$$

where the equality is by (3) with $i' = i-1$. Thus i's utility at any $a(i')$ with $i' < i$ is no larger than his utility at $a(i)$.

Similarly, for any $i' > i$, by Definition 5.5 we have

$$v_{i'a(i'-1)} - w_{a(i'-1)} = v_{i'a(i')} - w_{a(i')}. \tag{5}$$

Since $i < i'$ and $a(i'-1) \leq a(i')$, by Claim 5.1 we have $v_{i'a(i'-1)} - v_{i'a(i')} \leq v_{ia(i'-1)} - v_{ia(i')}$, that is,

$$v_{i'a(i'-1)} - v_{ia(i'-1)} \leq v_{i'a(i')} - v_{ia(i')}. \tag{6}$$

Subtracting corresponding sides of (6) from those of (5), we have

$$v_{ia(i'-1)} - w_{a(i'-1)} \geq v_{ia(i')} - w_{a(i')}.$$

Since this holds for all $i' > i$, we have

$$v_{ia(i)} - w_{a(i)} \geq v_{ia(i+1)} - w_{a(i+1)} \geq \cdots \geq v_{ia(n)} - w_{a(n)} \geq 0, \tag{7}$$

where the last inequality is further because $w_{a(n)} = 0$. Thus i's utility at any $a(i')$ with $i' > i$ is no larger than his utility at $a(i)$, and his utility at $a(i)$ is non-negative.

It remains to show that for any hospital $j \notin a([n])$, $v_{ia(i)} - w_{a(i)} \geq v_{ij} - w_j$. This is clearly true because $v_{ia(i)} - w_{a(i)} \geq 0$ and $v_{ij} - w_j = v_{ij} - \infty < 0$. Therefore A is stable and Lemma 5.6 holds. □

Lemma 5.6 implies that, for any ordered assignment function a, the set of stable assignments whose assignment function is a is non-empty. In the analysis below, for $S \subseteq [n]$, $a(S) \subseteq [m]$ is the image set of S under the function a.

LEMMA 5.7. *For any ordered assignment function a and any stable assignment $A = (a, w)$, A is optimal with respect to a if and only if it is tight at a.*

PROOF. We start by proving the "only if" part and let $A = (a, w)$ be a stable assignment that is optimal with respect to a.

First, assuming $w_{a(n)} > 0$, we define waiting vector w' as follows. For $j \in [m]$ such that there exists $i \in [n]$ with $a(i) = j$, let $w'_j = w_j - w_{a(n)}$. Note that $w'_j \geq 0$ by (2). For any other j, let $w'_j = w_j$. It is easy to see that the assignment $A' = (a, w')$ is still stable. However,

$$\begin{aligned} SW(A') \;&=\; \sum_{i\in[n]} v_{ia(i)} - w'_{a(i)} = \sum_{i\in[n]} \left(v_{ia(i)} - w_{a(i)} + w_{a(n)}\right) \\ &=\; SW(A) + nw_{a(n)} > SW(A), \end{aligned}$$

contradicting the optimality of A. Thus $w_{a(n)} = 0$.

Second, if $w_{a(i)} < v_{(i+1)a(i)} - v_{(i+1)a(i+1)} + w_{a(i+1)}$ for some $i < n$, then

$$v_{(i+1)a(i+1)} - w_{a(i+1)} < v_{(i+1)a(i)} - w_{a(i)}$$

and $a(i+1)$ does not maximize the utility of patient $i+1$, contradicting the stability of A. Thus for any $i < n$,

$$w_{a(i)} \geq v_{(i+1)a(i)} - v_{(i+1)a(i+1)} + w_{a(i+1)}.$$

To prove that the two sides of the above inequality are actually equal, assume for the sake of contradiction that for some $i < n$

$$w_{a(i)} > v_{(i+1)a(i)} - v_{(i+1)a(i+1)} + w_{a(i+1)}.$$

Letting $\delta = w_{a(i)} - (v_{(i+1)a(i)} - v_{(i+1)a(i+1)}) - w_{a(i+1)}$, we have $\delta > 0$. Since $a(i) \leq a(i+1)$, we have $v_{(i+1)a(i)} - v_{(i+1)a(i+1)} \geq 0$ and thus $\delta \leq w_{a(i)}$. Construct waiting vector w' as follows. For any hospital $j \in a(\{1, 2, \ldots, i\})$, let $w'_j = w_j - \delta$. Note that $0 \leq w'_j < w_j$. For any other j, let $w'_j = w_j$. Letting $A' = (a, w')$, we prove the following.

CLAIM 5.8. *A' is stable.*

PROOF. To begin with, all patients' utilities under A' are non-negative, as the waiting times only decrease from w. Next, we show that for any patient i', his utility in A' is still maximized at $a(i')$.

First of all, for any patient $i' \leq i$ and hospital j,

$$v_{i'a(i')} - w'_{a(i')} = v_{i'a(i')} - w_{a(i')} + \delta \geq v_{i'j} - w_j + \delta \geq v_{i'j} - w'_j, \tag{8}$$

where the second inequality is because $w'_j \geq w_j - \delta$. Accordingly, the utility of i' is maximized at $a(i')$ in A'.

Second of all, arbitrarily fix a patient $i' \geq i+1$. For any hospital $j \notin a(\{1, \ldots, i\})$, we have

$$v_{i'a(i')} - w'_{a(i')} \geq v_{i'a(i')} - w_{a(i')} \geq v_{i'j} - w_j = v_{i'j} - w'_j, \tag{9}$$

where the equality is because $w'_j = w_j$. It remains to consider the utility of patient i' in A' at an arbitrary hospital $j \in a(\{1, \ldots, i\})$.

Applying (8) to patient i and hospital j, we have

$$v_{ij} - w'_j \leq v_{ia(i)} - w'_{a(i)}. \tag{10}$$

Since $j = a(i'')$ for some $i'' \le i$ and a is ordered, we have $j \le a(i)$. Since $i < i'$, by Claim 5.1 we have $v_{ij} - v_{ia(i)} \ge v_{i'j} - v_{i'a(i)}$, that is,

$$v_{ij} - v_{i'j} \ge v_{ia(i)} - v_{i'a(i)}. \tag{11}$$

Subtracting corresponding sides of (11) from those of (10), we have

$$v_{i'j} - w'_j \le v_{i'a(i)} - w'_{a(i)}, \tag{12}$$

that is, *the utility of patient i' at j is no larger than his utility at $a(i)$ in A'*. Next, we show that *the utility of i' at $a(i)$ is no larger than his utility at $a(i')$ in A'*.

To do so, by definition we have

$$w'_{a(i)} = w_{a(i)} - \delta = v_{(i+1)a(i)} - v_{(i+1)a(i+1)} + w_{a(i+1)},$$

namely,

$$v_{(i+1)a(i)} - w'_{a(i)} = v_{(i+1)a(i+1)} - w_{a(i+1)}. \tag{13}$$

By the hypothesis,

$$v_{(i+1)a(i+1)} - w_{a(i+1)} > v_{(i+1)a(i)} - w_{a(i)},$$

implying $a(i+1) \neq a(i)$. Since a is ordered, $a(i+1) > a(i)$ and $a(\{1,\ldots,i\}) \subseteq \{1,\ldots,a(i)\}$. Thus

$$a(i+1) \notin a(\{1,\ldots,i\}) \quad \text{and} \quad w'_{a(i+1)} = w_{a(i+1)}, \tag{14}$$

where the second part together with (13) implies

$$v_{(i+1)a(i)} - w'_{a(i)} = v_{(i+1)a(i+1)} - w'_{a(i+1)}. \tag{15}$$

Since $i' \ge i+1$, and $a(i) \le a(i+1)$, again by Claim 5.1 we have

$$v_{(i+1)a(i)} - v_{i'a(i)} \ge v_{(i+1)a(i+1)} - v_{i'a(i+1)}. \tag{16}$$

Subtracting corresponding sides of (16) from those of (15), we have

$$v_{i'a(i)} - w'_{a(i)} \le v_{i'a(i+1)} - w'_{a(i+1)}.$$

Since $a(i+1) \notin a(\{1,\ldots,i\})$ by the first part of (14), applying (9) to hospital $a(i+1)$ we have

$$v_{i'a(i')} - w'_{a(i')} \ge v_{i'a(i+1)} - w'_{a(i+1)}.$$

Combining the two inequalities above, we have

$$v_{i'a(i')} - w'_{a(i')} \ge v_{i'a(i)} - w'_{a(i)}, \tag{17}$$

that is, the utility of i' at $a(i)$ is no larger than his utility at $a(i')$ in A', as desired. Combining (12) and (17), we have that for any $j \in a(\{1,\ldots,i\})$,

$$v_{i'a(i')} - w'_{a(i')} \ge v_{i'j} - w'_j. \tag{18}$$

Combining (8), (9) and (18), A' is stable and Claim 5.8 holds. $\square$

However,

$$\begin{aligned} SW(A') &= \sum_{i' \in [n]} v_{i'a(i')} - w'_{a(i')} \\ &= \sum_{i' \le i} \left[v_{i'a(i')} - w_{a(i')} + \delta\right] + \sum_{i' \ge i+1} \left[v_{i'a(i')} - w_{a(i')}\right] \\ &\ge SW(A) + \delta > SW(A), \end{aligned}$$

contradicting the fact that A is optimal with respect to a. Therefore the hypothesis is false and $w_{a(i)} = v_{(i+1)a(i)} - v_{(i+1)a(i+1)} + w_{a(i+1)}$ for any $i < n$, concluding the proof of the "only if" part.

Finally, to prove the "if" part, note that for any patient i, the tightness of A at a has uniquely pinned down $w_{a(i)}$ and thus the utility of i at $a(i)$. Accordingly, all assignments (whether stable or not) that have assignment function a and are tight at a have the same social welfare. Thus any such assignment that is stable must be optimal with respect to a, as desired. In sum, Lemma 5.7 holds. □

The following lemma shows that the social welfare of any stable assignment optimal with respect to a can be explicitly calculated from the patients' values.

LEMMA 5.9. *For any ordered assignment function a and any stable assignment $A = (a, w)$ optimal with respect to a, $SW(A) = \sum_{i<n} i(v_{ia(i)} - v_{(i+1)a(i)}) + nv_{na(n)}$.*

PROOF. For any $k < n$, let $U_k = \sum_{i=k}^{n-1} \left[v_{ia(i)} - w_{a(i)}\right]$. By Lemma 5.7, A is tight at a. Thus by Definition 5.5 we have

$$\begin{aligned} U_{n-1} &= v_{(n-1)a(n-1)} - w_{a(n-1)} = v_{(n-1)a(n-1)} - (v_{na(n-1)} - v_{na(n)}) - w_{a(n)} \\ &= v_{na(n)} + v_{(n-1)a(n-1)} - v_{na(n-1)}, \end{aligned}$$

and for any $k < n - 1$,

$$\begin{aligned} U_k &= \sum_{i=k}^{n-1} v_{ia(i)} - \sum_{i=k}^{n-1} w_{a(i)} = \sum_{i=k}^{n-1} v_{ia(i)} - \sum_{i=k}^{n-1} \left[v_{(i+1)a(i)} - v_{(i+1)a(i+1)} + w_{a(i+1)}\right] \\ &= \sum_{i=k}^{n-1} \left[v_{ia(i)} - v_{(i+1)a(i)}\right] + \sum_{i=k+1}^{n} v_{ia(i)} - \sum_{i=k+1}^{n} w_{a(i)} \\ &= v_{na(n)} + \sum_{i=k}^{n-1} \left[v_{ia(i)} - v_{(i+1)a(i)}\right] + \sum_{i=k+1}^{n-1} v_{ia(i)} - \sum_{i=k+1}^{n-1} w_{a(i)} \\ &= v_{na(n)} + \sum_{i=k}^{n-1} \left[v_{ia(i)} - v_{(i+1)a(i)}\right] + U_{k+1}. \end{aligned}$$

Thus

$$\begin{aligned} SW(A) &= \sum_{i\in[n]} \left[v_{ia(i)} - w_{a(i)}\right] = v_{na(n)} + U_1 = \cdots \\ &= nv_{na(n)} + \sum_{k=1}^{n-1} \sum_{i=k}^{n-1} \left[v_{ia(i)} - v_{(i+1)a(i)}\right] = nv_{na(n)} + \sum_{i=1}^{n-1} \sum_{k=1}^{i} \left[v_{ia(i)} - v_{(i+1)a(i)}\right] \\ &= \sum_{i=1}^{n-1} i(v_{ia(i)} - v_{(i+1)a(i)}) + nv_{na(n)}, \end{aligned}$$

and Lemma 5.9 holds. □

The above lemmas are the key tools for computing the optimal stable assignment and analyzing its complexity; see the discussion below.

5.2. The Complexity of Computing Optimal Stable Assignments

THEOREM 5.10. *It is NP-hard to compute an optimal stable assignment when the patients have proportional preferences.*

We prove Theorem 5.10 by applying the lemmas from Section 5.1 to proportional preferences. As mentioned in Section 3, for clarity, when the patients have proportional preferences, we explicitly represent the value v_{ij} as $v_i q_j$. Moreover, recall that we rename the patients and the hospitals so that

$$v_1 \geq v_2 \geq \cdots \geq v_n \quad \text{and} \quad q_1 \geq q_2 \geq \cdots \geq q_m. \tag{19}$$

PROOF OF THEOREM 5.10. Consider the decision version of the Provision-after-Wait problem:

$$\begin{aligned} DPaW \; = \; \{ & (q_1, \ldots, q_m, c_1, \ldots, c_m, v_1, \ldots, v_n, B, V) : \\ & \text{there exists a stable budget-feasible assignment } A \text{ s.t. } SW(A) \geq V\}. \end{aligned}$$

We show that $DPaW$ is NP-complete by a reduction from the Subset Sum problem:

$$SubsetSum = \{(s_1, \ldots, s_n, T) : \text{there exists } S \subseteq [n] \text{ such that } \sum_{i \in S} s_i = T\}.$$

Given an instance $\alpha = (s_1, \ldots, s_n, T)$ of $SubsetSum$, we assume without loss of generality that $s_1 \geq s_2 \geq \cdots \geq s_n$, and construct an instance $\gamma = (q_1, \ldots, q_m, c_1, \ldots, c_m, v_1, \ldots, v_n, B, V)$ of $DPaW$ as follows. Notice that we use the same symbol for both a variable and its binary representation, and the 1st bit refers to the rightmost bit.

— There are $m = 2n$ hospitals and n patients.
— For each $i \in [n]$, $q_i = c_i = s_i \cdot 2^{n(\lceil \log n \rceil + 1)} + 2^{(n-i)(\lceil \log n \rceil + 1)}$: q_i and c_i are obtained by appending $n(\lceil \log n \rceil + 1)$ bits of 0's to the right of the binary representation of s_i, and then setting the $(n-i)(\lceil \log n \rceil + 1) + 1$st bit to 1.
— For each $i \in [n]$, $q_{n+i} = c_{n+i} = 2^{(n-i)(\lceil \log n \rceil + 1)}$. That is, q_{n+i} and c_{n+i} consist of one bit of 1 followed by $(n-i)(\lceil \log n \rceil + 1)$ bits of 0's. Notice that the unique bit of 1 in q_{n+i} and c_{n+i} is aligned with the unique bit of 1 after s_i in q_i and c_i.
— $B = V = T \cdot 2^{n(\lceil \log n \rceil + 1)} + \sum_{i \in [n]} 2^{(n-i)(\lceil \log n \rceil + 1)}$. That is, B and V are obtained by appending $n(\lceil \log n \rceil + 1)$ bits of 0's to the right of the binary representation of T, and then set the $(n-i)(\lceil \log n \rceil + 1) + 1$st bit to 1 for each $i \in [n]$.
— For each $i \in [n]$, $v_i = \sum_{k=i}^{n} \frac{1}{k}$.

It is easy to see that the construction takes polynomial time and that γ satisfies (19). We have the following two lemmas.

LEMMA 5.11. *$\gamma \in DPaW \Rightarrow \alpha \in SubsetSum$.*

PROOF. Let $A = (a, w)$ be an optimal stable assignment of γ. By definition, A is optimal with respect to a. By Lemma 5.4 we assume without loss of generality that a is ordered. Thus by Lemma 5.9 we have $SW(A) = \sum_{i<n} i \cdot q_{a(i)} \cdot (v_i - v_{i+1}) + n \cdot q_{a(n)} \cdot v_n = \sum_{i<n} i \cdot q_{a(i)} \cdot \frac{1}{i} + n \cdot q_{a(n)} \cdot \frac{1}{n} = \sum_{i \in [n]} q_{a(i)}$. Since $q_j = c_j$ for any $j \in [m]$ and since A is budget feasible, $SW(A) = \sum_{i \in [n]} c_{a(i)} = C(A) \leq B = V$. Since $\gamma \in DPaW$, we have $SW(A) \geq V$ and thus $SW(A) = V$. In particular, for any $j \in [n]$, $SW(A)$ has a 1 at the $(n-j)(\lceil \log n \rceil + 1) + 1$st bit preceded by $\lceil \log n \rceil$ bits of 0's. We now show that for any $j \in [n]$,

$$|\{i \in [n] : a(i) \in \{j, n+j\}\}| = 1, \tag{20}$$

that is, there is exactly one patient assigned to either hospital j or hospital $n+j$.

To see why (20) is true, notice that for any $k \in [n]$ there are $\lceil \log n \rceil$ bits of 0's between the $(n-k+1)(\lceil \log n \rceil + 1) + 1$st bit and the $(n-k)(\lceil \log n \rceil + 1) + 1$st bit in the binary representation of any q_j. Since there are n patients, there is no carry to the $(n-k+1)(\lceil \log n \rceil + 1) + 1$st bit when computing $SW(A)$. Further notice that the only hospitals whose qualities contribute a 1 to the $(n-j)(\lceil \log n \rceil + 1) + 1$st bit of $SW(A)$ are hospitals j and $n+j$.

If more than one patients are assigned to either j or $n+j$, then the $\lceil \log n \rceil$ bits preceding the $(n-j)(\lceil \log n \rceil + 1) + 1$st bit of $SW(A)$ cannot be all 0's, and $SW(A) \neq V$. If no patient is assigned to either j or $n+j$, then the $(n-j)(\lceil \log n \rceil + 1) + 1$st bit of $SW(A)$ cannot be a 1, and again $SW(A) \neq V$. Thus there must be exactly one patient assigned to either hospital j or hospital $n+j$, and (20) holds.

By (20), the two sets $S = \{j \in [n] : j \in a(\{1, \ldots, n\})\}$ and $S' = \{j \in [n] : n+j \in a(\{1, \ldots, n\})\}$ form a partition of $[n]$, and $SW(A) = \sum_{i \in [n]} q_{a(i)} = \sum_{j \in S} q_j + \sum_{j \in S'} q_{n+j} = \sum_{j \in S} s_j \cdot 2^{n(\lceil \log n \rceil + 1)} + \sum_{j \in [n]} 2^{(n-j)(\lceil \log n \rceil + 1)}$. Since $SW(A) = V = T \cdot 2^{n(\lceil \log n \rceil + 1)} + \sum_{i \in [n]} 2^{(n-i)(\lceil \log n \rceil + 1)}$, we have $\sum_{j \in S} s_j = T$. Thus $\alpha \in SubsetSum$ and Lemma 5.11 holds. □

LEMMA 5.12. $\alpha \in SubsetSum \Rightarrow \gamma \in DPaW$.

PROOF. Since $\alpha \in SubsetSum$, there exists $S \subseteq [n]$ such that $\sum_{i \in S} s_i = T$. Let $k = |S|$ and $S = \{j_1, \ldots, j_k\}$, with $j_1 \leq j_2 \leq \cdots \leq j_k$. Further, let $S' = [n] \setminus S = \{j_{k+1}, \ldots, j_n\}$, with $j_{k+1} \leq j_{k+2} \leq \cdots \leq j_n$. We construct an assignment $A = (a, w)$ as follows.

— $a(i) = j_i$ for any $i \leq k$, and $a(i) = n + j_i$ for any $i \geq k+1$.
— $w_{a(n)} = 0$, $w_{a(i)} = (q_{a(i)} - q_{a(i+1)})v_{i+1} + w_{a(i+1)}$ for any $i < n$, and $w_j = v_1 q_1$ for any $j \notin a(\{1, \ldots, n\})$.

Notice that $a(1) \leq a(2) \leq \cdots \leq a(n) = n + j_n \leq m$. Thus a is a well defined function from $[n]$ to $[m]$ and is ordered. Also notice that A is tight at a.

The cost of A is $C(A) = \sum_{i \in [n]} c_{a(i)} = \sum_{i \leq k} c_{j_i}$
$+ \sum_{i \geq k+1} c_{n+j_i} = T \cdot 2^{n(\lceil \log n \rceil + 1)} + \sum_{j \in [n]} 2^{(n-j)(\lceil \log n \rceil + 1)} = B$. Thus A is budget feasible.

Since A is tight at a, by Lemma 5.6, A is stable. By Lemma 5.7, A is optimal with respect to a. Thus $SW(A) = \sum_{i=1}^{n-1} i \cdot q_{a(i)} \cdot (v_i - v_{i+1}) + n \cdot q_{a(n)} \cdot v_n = \sum_{i=1}^{n-1} i \cdot q_{a(i)} \cdot \frac{1}{i} + n \cdot q_{a(n)} \cdot \frac{1}{n} = \sum_{i \in [n]} q_{a(i)} = \sum_{i \in [n]} c_{a(i)} = C(A) = B = V$. Therefore A is a stable budget feasible assignment with $SW(A) \geq V$. Accordingly, $\gamma \in DPaW$ and Lemma 5.12 holds. □

By Lemmas 5.11 and 5.12, $\alpha \in SubsetSum$ if and only if $\gamma \in DPaW$. Thus, $DPaW$ is NP-complete and Theorem 5.10 holds. □

COROLLARY 5.13. *It is NP-hard to compute an optimal stable assignment when the patients are d-ordered.*

5.3. An FPTAS for the Optimal Stable Assignments

Next, we show that there is an efficient algorithm that produces a stable assignment with social welfare arbitrarily close to the optimum for d-ordered preferences. Letting A^{opt} be an optimal stable assignment, we have the following.

THEOREM 5.14. *When the patients are d-ordered, there exists a fully polynomial-time approximation scheme (FPTAS) for the optimal stable assignments. Given any $\epsilon > 0$, the FPTAS runs in time $O((n+m)n^3 m/\epsilon)$ and outputs a stable budget-feasible assignment $A = (a, w)$ with $SW(A) \geq (1-\epsilon)SW(A^{opt})$.*

We prove Theorem 5.14 in Section 5.4. Below we introduce the key ideas and ingredients of the proof. By Lemma 5.9, for any ordered assignment function a we can define the *social welfare of* a, $SW(a)$, to be the social welfare of stable assignments optimal with respect to a. That is,

$$SW(a) = \sum_{i<n} i(v_{ia(i)} - v_{(i+1)a(i)}) + nv_{na(n)}. \quad (21)$$

An assignment function a is ***budget-feasible*** if $C(a) = \sum_{i\in[n]} c_{a(i)} \leq B$.

Definition 5.15. An ordered assignment function a is *optimal* if

$$a \in \operatorname*{argmax}_{a' \text{is ordered and budget-feasible}} SW(a').$$

Given an ordered assignment function a, by Lemmas 5.6 and 5.7 we can construct, in time $O(m+n)$, a stable assignment A optimal with respect to a: that is, the tight assignment in Definition 5.5. If a is optimal, then A is an optimal stable assignment. Thus, to prove Theorem 5.14 it suffices to approximate the optimal ordered assignment function.

If there exists a hospital j such that $c_j < c_{j+1}$, then for any ordered assignment function a and for all patients assigned to hospital $j+1$, by reassigning them to j we get another ordered assignment function a' such that $C(a') \leq C(a)$ and $SW(a') \geq SW(a)$. Accordingly, we can focus on ordered assignment functions that do not assign any patient to $j+1$. That is, we can assume without loss of generality that

$$c_1 \geq c_2 \geq \cdots \geq c_m.$$

Moreover, as mentioned in Remark 3.4, for arbitrary d-ordered preferences $v = (v_{ij})_{i\in[n],j\in[m]}$, v may not satisfy the additional ordering property that $v_{1j} \geq v_{2j} \geq \cdots \geq v_{nj}$ for each hospital j. However, since $v_{ij} = v^d_{i(j+1)} + \cdots + v^d_{im} + v_{im}$ for any patient i and hospital j, if we define $v'_{ij} = v_{ij} - v_{im}$, then the resulting preferences $v' = (v'_{ij})_{i\in[n],j\in[m]}$ keep the original differences and do satisfy the additional ordering property. Also, $v'_{im} = 0$ for any patient i. Note that an optimal stable assignment for v is also an optimal stable assignment for v', and vice versa. Indeed, given a waiting vector and an assignment, whether a patient wants to deviate or not only depends on the differences of his values for different hospitals, not the actual values; and the social welfare under v and that under v' always differ by $\sum_{i\in[n]} v_{im}$. We also need to argue that the optimal stable assignment $A = (a, w)$ for v gives a non-negative utility for every patient i under v': indeed, $w_{a(n)} = 0$ by Definition 5.5 and Lemmas 5.6 and 5.7, thus $v_{ia(i)} - w_{a(i)} \geq v_{ia(n)} - w_{a(n)} = v_{ia(n)} \geq v_{im}$ and $v'_{ia(i)} - w_{a(i)} = v_{ia(i)} - v_{im} - w_{a(i)} \geq 0$. By approximating the optimal ordered assignment function for v', we immediately get an approximation to the optimal ordered assignment function for v. That is, without loss of generality, we focus on d-ordered preferences where $v_{im} = 0$ for any patient i and

$$v_{1j} \geq v_{2j} \geq \cdots \geq v_{nj} \quad (22)$$

for any hospital j, Note that this class still contains proportional preferences as special cases.

Below we define a new combinatorial optimization problem and construct an FPTAS for it, which will give us an FPTAS for the optimal ordered assignment function.

Definition 5.16. The *ordered Knapsack* problem has m items, n players, and a budget B. Each item j has n copies, with cost $c_j \geq 0$ each. Each player i has value $u_{ij} \geq 0$ for item j. We have $c_1 \geq c_2 \geq \cdots \geq c_m$, $u_{i1} \geq u_{i2} \geq \cdots \geq u_{im}$ for each $i \in [n]$, and $nc_m \leq B < nc_1$. An *assignment* is a function $a : [n] \to [m]$ such that

$a(1) \leq a(2) \leq \cdots \leq a(n)$. The *social welfare* of a is $SW(a) = \sum_{i\in[n]} u_{ia(i)}$, and the *cost* is $C(a) = \sum_{i\in[n]} c_{a(i)}$. The goal is to find an assignment with cost no larger than B and the maximum social welfare.

Intuitively, the ordered-Knapsack problem has a knapsack where the order of the items packed in it affects their values —the "players" can be considered as ordered slots in the knapsack.[7] By (21) and (22), we can reduce the problem of finding the optimal ordered assignment function to the ordered Knapsack problem by taking, for any $j \in [m]$,

$$u_{ij} = i(v_{ij} - v_{(i+1)j}) \geq 0$$

for any $i < n$, and $u_{nj} = nv_{nj}$. Since the patients are d-ordered, it is easy to verify that $u_{i1} \geq u_{i2} \geq \cdots \geq u_{im}$ for each $i \in [n]$. Any assignment of the resulting ordered Knapsack problem is an ordered assignment function of the original Provision-after-Wait problem, with the same cost and social welfare. Thus, letting a^{opt} be the optimal assignment for the ordered Knapsack problem, to prove Theorem 5.14 it suffices to construct an FPTAS for a^{opt}.

THEOREM 5.17. *There exists a fully polynomial-time approximation scheme (FPTAS) for the ordered Knapsack problem such that, given any $\epsilon > 0$, it runs in time $O((n+m)n^3m/\epsilon)$ and outputs an assignment a with $C(a) \leq B$ and $SW(a) \geq (1-\epsilon)SW(a^{opt})$.*

5.4. Proving Theorems 5.14 and 5.17

According to the discussion above, Theorem 5.14 depends on Theorem 5.17 and we prove Theorem 5.17 first. To do so, we first derive some lower/upper-bounds for $SW(a^{opt})$. For any player i, letting j_i be the smallest item that i can be assigned to in any budget feasible assignment, we have

$$j_i = \min\{j \in [m] : ic_j + (n-i)c_m \leq B\}.$$

Indeed, if i is assigned to some item $j' < j_i$, then by definition the minimum cost of such assignments is achieved by assigning players $1, \ldots, i$ to item j' and all others to item m, leading to cost $ic_{j'} + (n-i)c_m > B$ by the definition of j_i. Notice that j_i is always well defined, as assigning all players to item m is budget feasible. For each $i \in [n]$, let a^i be the assignment which assigns players $1, \ldots, i$ to item j_i and all others to item m. We have that all the a^i's are budget feasible and $a^i(1) \leq \cdots \leq a^i(n)$. Thus for each $i \in [n]$, by definition we have

$$SW(a^{opt}) \geq SW(a^i) \geq u_{ij_i}.$$

Also, by the definition of the j_i's we have $a^{opt}(i) \geq j_i$, and thus for any i,

$$u_{ia^{opt}(i)} \leq u_{ij_i}.$$

Accordingly, letting $V = \max_{i\in[n]} u_{ij_i}$, we have

$$nV \geq \sum_i u_{ij_i} \geq \sum_i u_{ia^{opt}(i)} = SW(a^{opt}) \geq V. \tag{23}$$

[7]Such a scenario widely exists in real life. For example, in school choices the order may represent the priority of being admitted to different schools. Indeed, priority list has been widely studied in the Economics literature (see, e.g., [Bogomolnaia and Moulin 2001; Abdulkadiroğlu et al. 2005; Budish et al. 2013]). But the model and the concerns there are different from ours, e.g., the optimization goal is usually not utilitarian, and there is no budget constraints. Thus we do not elaborate on this line of research. Also notice that the ordered Knapsack problem is quite different from the partially ordered Knapsack problem studied in [Kolliopoulos and Steiner 2007]. In the latter each item has a fixed value and the outcome is a set instead of a function from players to items.

The following lemma shows the existence of a pseudo-polynomial time algorithm for ordered Knapsack.

LEMMA 5.18. *There exists a dynamic program that runs in time $O((n+m)n^2mV)$ and computes an optimal assignment for the ordered Knapsack problem.*

PROOF. For any assignment a and player i, let

$$SW(a,i) = \sum_{i'=i}^{n} u_{i'a(i')}$$

be the contribution of players $i,\dots,n$ to $SW(a)$. For any $i \in [n]$, $j \in [m]$, and $s \in \{0,1,\dots,nV\}$, we are interested in the *minimum cost*, denoted by $C(i)(j)(s)$, needed for players $i,\dots,n$ to make contribution s to the social welfare, when player i is assigned to item j. More precisely, letting

$$SW(i,j) = \sum_{i'=i}^{n} u_{i'j}$$

be the contribution of players $i,\dots,n$ when they are all assigned to j,

$$C(i)(j)(s) = \begin{cases} c_j + \min\limits_{\substack{a: j=a(i)\le a(i+1)\le\cdots\le a(n),\\ SW(a,i)\ge s}} \sum_{i'>i} c_{a(i')} & \text{if } SW(i,j)\ge s,\\ +\infty & \text{otherwise.}\end{cases} \tag{24}$$

Notice that $C(i)(j)(s) = +\infty$ means it is impossible for players $i,\dots,n$ to make contribution s to the social welfare even if all of them are assigned to j, and thus impossible to make such contribution at j and items after j. In practice, $+\infty$ can be replaced by $B+1$ (or any number larger than B and of polynomial length).

Also notice that, for any $s \le nV$, $\min_{j\in[m]} C(1)(j)(s)$ is the minimum cost of any assignment whose social welfare is at least s. Thus we immediately have the following claim, of which the proof is omitted.

CLAIM 5.19. *For any optimal assignment a,*

$$SW(a) = \max\{s : \min_{j\in[m]} C(1)(j)(s) \le B\}.$$

In order to compute the $C(i)(j)(s)$'s, we prove the following.

CLAIM 5.20. *$C(n)(j)(s) = c_j$ for any $j \in [m]$ and $s \le u_{nj}$; $C(i)(j)(0) = c_j + (n-i)c_m$ for any $i < n$ and $j \in [m]$; and for any $i < n$, $j \in [m]$ and $0 < s \le SW(i,j)$,*

$$C(i)(j)(s) = c_j + \min_{j'\ge j} C(i+1)(j')(\max\{s-u_{ij},0\}). \tag{25}$$

Finally, $C(i)(j)(s) = +\infty$ in all other cases.

PROOF. We only prove (refequ:C-1), since other equalities follow directly from the definition of the $C(i)(j)(s)$'s. Notice that for any assignment a with $a(i) = j$, $SW(a,i) \ge s$ if and only if $SW(a,i+1) \ge \max\{s-u_{ij},0\}$. For any $j' \ge j$, let

$$S_{j'} = \{a : j = a(i), j' = a(i+1) \le \cdots \le a(n), SW(a,i+1) \ge \max\{s-u_{ij},0\}\}.$$

We have

$$\{a : j = a(i) \le a(i+1) \le \cdots \le a(n), SW(a,i) \ge s\} = \cup_{j'\ge j} S_{j'}$$

and

$$C(i)(j)(s) = c_j + \min_{\substack{a:j=a(i)\leq a(i+1)\leq\cdots\leq a(n),\\ SW(a,i)\geq s}} \sum_{i'>i} c_{a(i')} = c_j + \min_{j'\geq j} \min_{a\in S_{j'}} \left(c_{j'} + \sum_{i'>i+1} c_{a(i')} \right), \tag{26}$$

where $\min\limits_{a\in S_{j'}} \left(c_{j'} + \sum_{i'>i+1} c_{a(i')}\right)$ is defined to be $+\infty$ whenever $S_{j'} = \emptyset$. Notice that $S_{j'} = \emptyset$ implies $SW(i+1, j') < \max\{s - u_{ij}, 0\}$, and thus by (24) we have

$$C(i+1)(j')(\max\{s-u_{ij},0\}) = +\infty = \min_{a\in S_{j'}} \left(c_{j'} + \sum_{i'>i+1} c_{a(i')} \right). \tag{27}$$

Also notice that $S_{j'} \neq \emptyset$ for some j'. In fact, $s \leq SW(i,j)$ implies $SW(i+1,j) \geq \max\{s-u_{ij},0\}$, and thus $S_j \neq \emptyset$. For any $S_{j'} \neq \emptyset$, we have $SW(i+1,j') \geq \max\{s-u_{ij},0\}$, and

$$\begin{aligned} C(i+1)(j')(\max\{s-u_{ij},0\}) &= c_{j'} + \min_{\substack{a:j'=a(i+1)\leq\cdots\leq a(n),\\ SW(a,i+1)\geq\max\{s-u_{ij},0\}}} \sum_{i'>i+1} c_{a(i')} \\ &= \min_{a\in S_{j'}} \left(c_{j'} + \sum_{i'>i+1} c_{a(i')} \right), \end{aligned} \tag{28}$$

where the second equality is because that, given $a(i+1) = j' \geq j = a(i)$, neither $SW(a, i+1)$ nor $\sum_{i'>i+1} c_{a(i')}$ depends on $a(i)$.

Combining (26), (27) and (28), we have

$$C(i)(j)(s) = c_j + \min_{j'\geq j} C(i+1)(j')(\max\{s-u_{ij},0\}),$$

and Claim 5.20 holds. □

Equation (25) immediately leads to a dynamic program computing all $C(i)(j)(s)$'s, with other equations in Claim 5.20 as initialization conditions. Since it takes $O(n)$ time to compute each $SW(i,j)$, by Claim 5.20 it takes $O(n+m)$ time to compute each $C(i)(j)(s)$ given the $C(i+1)(j')(s')$'s. Thus the dynamic program takes space $O(n^2mV)$ and runs in time $O((n+m)n^2mV)$. By Claim 5.19, given the $C(i)(j)(s)$'s, the social welfare of the optimal assignment can be computed in time $O(mnV)$.

Moreover, the dynamic program can keep track of the optimal j''s when computing the $C(i)(j)(s)$'s according to (25). Once the $C(1)(j)(s)$ corresponding to the optimal social welfare is found, the dynamic program can trace back the stored j''s and recover the assigned item for each player, and thus compute the corresponding optimal assignment. The total space is still $O(n^2mV)$ and the running time is still $O((n+m)n^2mV)$. Therefore Lemma 5.18 holds. □

For completeness, we provide the dynamic program in Algorithm 1, where for each $i < n$, $j \in [m]$ and $s \leq nV$, $\hat{a}(i)(j)(s)$ represents the item to which player $i+1$ is assigned to, in order for players $i,\ldots,n$ to make contribution s at cost $C(i)(j)(s)$. By scaling the players' values and running the dynamic program on the scaled input, we obtain an FPTAS for the ordered Knapsack problem, see below.

PROOF OF THEOREM 5.17.. Given $c_1,\ldots,c_m$, $u_{11},\ldots,u_{nm}$, B and $\epsilon > 0$, our algorithm OKNAPSACK works as follows. Let the j_i's and V be defined as before, $K = \frac{\epsilon V}{n}$, and $u'_{ij} = \lfloor \frac{u_{ij}}{K} \rfloor$ for any $i \in [n]$ and $j \in [m]$. Run Algorithm 2 with input $(c_1,\ldots,c_m,u'_{11},\ldots,u'_{nm},B)$ and return its output a.

ALGORITHM 1: A Dynamic Program for Ordered Knapsack

Input : cost c_j for each $j \in [m]$, value u_{ij} for each $i \in [n]$ and $j \in [m]$, and budget B.
Output: an optimal assignment a.

Initialization:
for i *from* 1 *to* n **do**
  $j_i = \min\{j \in [m] : ic_j + (n-i)c_m \leq B\}$;
end for
$V = \max_{i \in [n]} u_{ij_i}$;
for j *from* 1 *to* m *and* s *from* 0 *to* nV **do**
  if $s \leq u_{nj}$ **then**
    $C(n)(j)(s) = c_j$;
  else
    $C(n)(j)(s) = B + 1$;
  end if
end for
for i *from* 1 *to* $n-1$ *and* j *from* 1 *to* m **do**
  $C(i)(j)(0) = c_j + (n-i)c_m$; $\hat{a}(i)(j)(0) = m$;
end for

Compute $C(i)(j)(s)$ and $\hat{a}(i)(j)(s)$:
for i *from* $n-1$ *to* 1 **do**
  for j *from* 1 *to* m **do**
    $SW(i,j) = \sum_{i'=i}^{n} u_{i'j}$;
    for s *from* 1 *to* nV **do**
      if $s \leq SW(i,j)$ **then**
        $\hat{j} = \operatorname{argmin}_{j' \geq j} C(i+1)(j')(\max\{s - u_{ij}, 0\})$, with ties broken lexicographically;
        $C(i)(j)(s) = c_j + C(i+1)(\hat{j})(\max\{s - u_{ij}, 0\})$;
        $\hat{a}(i)(j)(s) = \hat{j}$;
      else
        $C(i)(j)(s) = B + 1$; (It doesn't matter what $\hat{a}(i)(j)(s)$ is in this case.)
      end if
    end for
  end for
end for

Compute a:
for s *from* nV *to* 0 **do**
  $\hat{j} = \operatorname{argmin}_{j \in [m]} C(1)(j)(s)$, with ties broken lexicographically;
  if $C(1)(\hat{j})(s) \leq B$ **then**
    $a(1) = \hat{j}$; **break**;
  end if
end for
for i *from* 1 *to* $n-1$ **do**
  $a(i+1) = \hat{a}(i)(\hat{j})(s)$; $s = \max\{s - u_{i\hat{j}}, 0\}$; $\hat{j} = a(i+1)$;
end for
return a

Since $u_{i1} \geq \cdots \geq u_{im}$ for any $i \in [n]$, we have $u'_{i1} \geq \cdots \geq u'_{im}$ for any i, and thus the input to the dynamic program is a valid instance of ordered Knapsack. Since the budget and the costs do not change, the j_i's computed on the scaled input are still the same as before. Let $V' = \max_{i \in [n]} u'_{ij_i}$ be the counterpart of V for the scaled input. We have $V' \leq \max_{i \in [n]} \frac{u_{ij_i}}{K} = \frac{V}{K} = \frac{n}{\epsilon}$. Thus the dynamic program runs in time $O((n+m)n^2mV') = O((n+m)n^3m/\epsilon)$, and so does the algorithm oKNAPSACK.

Below we analyze the approximation ratio. For any assignment a', let $SW(a')$ and $SW'(a')$ respectively be the social welfare of a' in the original ordered Knapsack problem and in the scaled input to the dynamic program. We have

$$\begin{aligned} SW(a) &= \sum_{i \in [n]} u_{ia(i)} \geq K \sum_{i \in [n]} u'_{ia(i)} = K \cdot SW'(a) \geq K \cdot SW'(a^{opt}) = K \sum_{i \in [n]} u'_{ia^{opt}(i)} \\ &\geq K \sum_{i \in [n]} \Big(\frac{u_{ia^{opt}(i)}}{K} - 1 \Big) = \sum_{i \in [n]} u_{ia^{opt}(i)} - nK = SW(a^{opt}) - \epsilon V \\ &\geq SW(a^{opt}) - \epsilon SW(a^{opt}) = (1-\epsilon) SW(a^{opt}), \end{aligned}$$

where the first and the third inequalities are by the definition of u'_{ij}'s, the second is because a is optimal under the scaled input, and the last is by (23).

In sum, Theorem 5.17 holds. □

Finally we prove Theorem 5.14.

PROOF OF THEOREM 5.14. It is easy to see that algorithm oKNAPSACK constructed in the proof of Theorem 5.17 can be applied to the Provision-after-Wait problem. Indeed, given an instance $\gamma = ((v_{ij})_{i \in [n], j \in [m]}, (c_j)_{j \in [m]}, B)$ of the Provision-after-Wait problem with d-ordered preferences, for any $j \in [m]$, we can take

$$u_{ij} = i(v_{ij} - v_{(i+1)j}) \text{ for any } i < n \text{ and } u_{nj} = nv_{nj}.$$

Then $\kappa = (c_1, \ldots, c_m, u_{11}, \ldots, u_{nm}, B)$ is a valid instance of ordered Knapsack. Moreover, any assignment a of κ is an ordered assignment function of γ with the same cost and the same social welfare, and vice versa. Thus a^{opt} is an optimal assignment function for γ and $SW(A^{opt}) = SW(a^{opt})$. By Theorem 5.17, the assignment a output by oKNAPSACK with input κ is budget-feasible and $SW(a) \geq (1-\epsilon)SW(a^{opt})$. Letting $A = (a, w)$ be the tight assignment at a as in Definition 5.5, we have that A is stable, budget-feasible, and optimal with respect to a, by Lemmas 5.6 and 5.7. Thus $SW(A) = SW(a) \geq (1-\epsilon)SW(a^{opt}) = (1-\epsilon)SW(A^{opt})$.

It takes $O(mn)$ time to construct κ from γ, and $O(n+m)$ time to construct A from a. Thus in total A can be computed in time $O(mn + (n+m)n^3m/\epsilon + n + m) = O((n+m)n^3m/\epsilon)$, and Theorem 5.14 holds. □

6. ASSIGNMENTS USING LOTTERIES

Lotteries have been widely used in school choices, but so far we are not aware of any use of lotteries in healthcare. Given the welfare-burning effect of waiting times, it is interesting and important to understand the performances of different rationing tools. If the planner is allowed to ask the patients to enter lotteries, the space of possible assignment schemes is much larger and presumably better social welfare can be obtained. In this section, we consider optimal lottery schemes for d-ordered and proportional preferences respectively.

Different from results in previous sections where the set of patients is discrete, the results in this section are provided for a continuous population of patients. We will also reverse the orders of the hospitals and the patients. As will become clear later, these changes in the model not only make the statement and the analysis more succinct, but also let us discover an alternative interpretation of our results from a different but very natural aspect. For completeness, in Section 6.3 we also discuss the counterparts of our results for discrete patients.

The Continuous Model. Formally speaking, the patients are indexed by the interval $[0, 1]$ and, for each hospital $j \in [m]$, the *valuation function* $v_j : [0, 1] \to \mathbb{R}^+$ specifies the value of each patient $x \in [0, 1]$ for hospital j. For patients with common preferences, we reverse the order of the hospitals so that $v_1(x) \leq v_2(x) \leq \cdots \leq v_m(x)$ for each patient x and $c_1 \leq c_2 \leq \cdots \leq c_m$. For d-ordered preferences, we also reverse the order of the patients so that $v_j(x) - v_{j-1}(x) \geq v_j(x') - v_{j-1}(x')$ for any hospital $j \in \{2, \ldots, m\}$ and patients $x' < x$. By Remark 3.4, we assume without loss of generality that each function v_j is non-decreasing. Also, by adding a dummy patient, we assume $v_j(0) = 0$ for each hospital j.

Since the key factors affecting the patients' choices are not the actual values but the differences of each patient's values at different hospitals, we can redefine the valuation functions starting with the differences. More precisely, for each hospital j, let $f_j : [0, 1] \to \mathbb{R}^+$ be a function that is non-decreasing and $f_j(0) = 0$. We consider the patients' values as $v_j(x) = \sum_{k=1}^{j} f_k(x)$ for any $x \in [0, 1]$ —note that the resulting patients are indeed d-ordered. The f_j's are referred to as the *difference functions*. For proportional preferences with $q_1 \leq q_2 \leq \cdots \leq q_m$, only one valuation function $v(x)$ is needed and $v_j(x) = q_j v(x)$ for each hospital j. Moreover, $f_1(x) = q_1 v(x)$ and $f_j(x) = (q_j - q_{j-1})v(x)$ for any $j \in \{2, \ldots, m\}$.

For simplicity, we assume that each f_j (and thus $v(x)$ for proportional preferences) is strictly increasing and twice differentiable. Our approach works as long as each f_j is non-decreasing and piecewise twice differentiable, but in this more general model the analysis is unnecessarily complicated without bringing in new insights.

Lottery Schemes. Below we define lotteries and lottery schemes for the Provision-after-Wait problem.

Definition 6.1. A *lottery* λ is a tuple of non-negative reals, $\lambda = (p_1, \ldots, p_m, w)$, such that $\sum_{j \in [m]} p_j \leq 1$. A *lottery scheme* L is a set of lotteries such that there exists $\lambda = (p_1, \ldots, p_m, w) \in L$ with $w = 0$.

Given a lottery scheme L, a patient x choosing a lottery $\lambda = (p_1, \ldots, p_m, w) \in L$ waits for time w and then gets assigned to each hospital j with probability p_j. Patient x's *(expected) utility* under λ is $u(x, \lambda) = (\sum_{j \in [m]} p_j v_j(x)) - w$.[8] Each patient chooses a lottery to maximize his utility. That is, denoting by $\lambda^L(x) = (p_1^L(x), \ldots, p_m^L(x), w^L(x)) \in L$ the choice of patient x given a lottery scheme L, we have that for any $\lambda \in L$,

$$u(x, \lambda^L(x)) \geq u(x, \lambda). \tag{29}$$

Since a lottery scheme includes a lottery with waiting time 0, we have $u(0, \lambda^L(0)) = 0$ and $u(x, \lambda^L(x)) \geq 0$ for any x.

In reality, a lottery scheme represents a set of healthcare options the patients can choose from: some lotteries may have short waiting times but the probability of going to the more preferred hospitals is low; while others may have long waiting times but then with high probability the patient will be assigned to a top hospital.

Note that for any two lotteries $\lambda_1, \lambda_2 \in L$, any convex combination $\alpha\lambda_1 + (1 - \alpha)\lambda_2$ can be realized by a patient choosing λ_1 with probability α and λ_2 with probability

[8] If $\sum_j p_j < 1$ then with probability $1 - \sum_j p_j$ the patient does not get served. If each patient has to be served, then we simply require $\sum_j p_j = 1$ and our results still hold.

$1 - \alpha$. Thus, if the patients randomize perfectly then without loss of generality we can assume L is convex. But we do not need the convexity condition in our analysis.

Since the patients are infinitesimal, each hospital j's cost c_j denotes the cost for serving 1 unit of the population at j. The *(expected) cost* of L is $C(L) = \int_0^1 \sum_{j\in[m]} p_j^L(x)c_j dx$ and L is *budget-feasible* if $C(L) \leq B$. Similar to the discrete case, we assume the budget is not enough to serve the whole population at the most expensive hospital, but is enough at the cheapest hospital: that is, $\min_{j\in[m]} c_j \leq B < \max_{j\in[m]} c_j$. Thus there always exists a budget-feasible lottery scheme: sending all patients to the cheapest hospital with probability 1. The *(expected) social welfare* of L is $SW(L) = \int_0^1 u(x, \lambda^L(x))dx$. We denote by L^{opt} the *optimal lottery scheme*:

$$L^{opt} \in \operatorname*{argmax}_{L \text{ is budget-feasible}} SW(L).$$

For each $x \in [0,1]$, we denote by $\lambda^{opt}(x) = (p_1^{opt}(x), \ldots, p_m^{opt}(x), w^{opt}(x))$ the choice of patient x under L^{opt}.

A *stable assignment* in the continuous model, $A = (a, w)$, is defined as before, except that a is a function from $[0,1]$ to $[m]$. It is easy to see that A is equivalent to the following lottery scheme $L = \{\lambda_1, \ldots, \lambda_m\}$: for each $j \in [m]$, $\lambda_j = (p_{j1}, \ldots, p_{jm}, w_j)$, where $p_{jj} = 1$ and $p_{jj'} = 0$ for any $j' \neq j$. Given L, each patient x chooses $\lambda_{a(x)}$, which corresponds to being assigned to $a(x)$ with probability 1 after waiting $w_{a(x)}$. Thus we have

$$SW(L^{opt}) \geq SW(A^{opt}),$$

where A^{opt} is the optimal stable assignment.

Note that a lottery scheme in general is "stable in expectation" but not "ex-post stable": that is, a patient's chosen lottery maximizes his expected utility, but the sampled hospital may not be the one he values the most.

Randomized Assignments. Besides stable assignments, another class of lottery schemes is of particular interest: that consists of a single lottery with waiting time 0. Such a lottery scheme does not give the patients any choice and randomly assigns each one of them to hospitals based on pre-specified probabilities. See the definition below.

Definition 6.2. A *randomized assignment* R is a tuple of non-negative reals, $R = (p_1, \ldots, p_m)$, such that $\sum_{j\in[m]} p_j \leq 1$.

According to R, each patient is assigned to each hospital j with probability p_j and without waiting. The (expected) social welfare of R is $SW(R) = \int_0^1 \sum_{j\in[m]} p_j v_j(x)dx$, and the (expected) cost is $C(R) = \sum_{j\in[m]} p_j c_j$. Note that sending all patients to the cheapest hospital with probability 1 is a budget-feasible randomized assignment. We denote by R^{opt} the *optimal randomized assignment*, that is,

$$R^{opt} \in \operatorname*{argmax}_{R \text{ is budget-feasible}} SW(R).$$

As a randomized assignment is a special lottery scheme, $SW(L^{opt}) \geq SW(R^{opt})$.

6.1. Randomized v.s. Stable Assignments for d-Ordered Preferences

Next, we compare the optimal randomized assignments and the optimal stable assignments when the patients are d-ordered. Note that the former does not have any waiting time but does not give the patients any choice; while the latter allows the patients to freely choose hospitals but burns some social welfare by using waiting times to balance supply and demand. These two types of assignment schemes separate randomness from waiting time and allow us to compare their individual performances.

Roughly speaking, the advantage of randomized assignments comes when the patients are d-ordered but not "super d-ordered": in other words, the difference functions $f_1, \ldots, f_m$ are increasing but do not increase too fast. In particular, if the difference functions increase extremely fast, then the patients close to 1 have values so high that, by assigning them deterministically to the best hospital and all the others to the cheapest hospital to meet the budget constraint, the planner generates a lot of social welfare in a stable assignment solely from patients with high values. On the other hand, randomized assignments cannot assign all patients to the best hospital with probability 1 due to the budget constraint, and any randomized assignment that do not assign patients with high values to the best hospital with probability 1 lose a lot of social welfare from them.

To properly formalize the intuition above, we say that a non-decreasing function $f : [0, 1] \to \mathbb{R}^+$ is *almost concave* if the function

$$g(x) \triangleq (1 - x)f'(x)$$

is non-increasing. Note that (1) f is almost concave if and only if $g'(x) \leq 0$, that is, $f''(x) \leq \frac{f'(x)}{1-x}$; and (2) if f is concave then it is almost concave. Indeed, $g'(x) = -f'(x) + (1 - x)f''(x)$ and, if $f(x)$ is concave then $f''(x) \leq 0 \leq \frac{f'(x)}{1-x}$, as $f'(x) \geq 0$ and $x \in [0, 1]$. Non-decreasing concave functions are natural examples for functions that do not increase too fast: that is why we choose the term "almost concave". We have the following.

THEOREM 6.3. *For d-ordered preferences, if $f_j(x)$ is almost concave for every $j \in [m]$, then $SW(R^{opt}) \geq SW(A^{opt})$.*

In Section 6.3 we provide a simple example where the f_j's are not almost concave and the optimal stable assignment does better. To prove Theorem 6.3, we start with the following claim.

CLAIM 6.4. *For any stable assignment $A = (a, w)$ and $x, x' \in [0, 1]$ with $x < x'$, we have $a(x) \leq a(x')$ and $w_{a(x)} \leq w_{a(x')}$.*

PROOF. By the definition of stable assignments, we have

$$v_{a(x)}(x) - w_{a(x)} \geq v_{a(x')}(x) - w_{a(x')} \quad \text{and} \quad v_{a(x')}(x') - w_{a(x')} \geq v_{a(x)}(x') - w_{a(x)}.$$

Summing up the two inequalities side by side and rearranging terms, we have

$$v_{a(x')}(x') - v_{a(x')}(x) \geq v_{a(x)}(x') - v_{a(x)}(x),$$

that is

$$\sum_{k=1}^{a(x')} [f_k(x') - f_k(x)] \geq \sum_{k=1}^{a(x)} [f_k(x') - f_k(x)].$$

Notice that $f_k(x') - f_k(x) > 0$ for any $k \in [m]$, since f_k is strictly increasing and $x < x'$. Thus $a(x) \leq a(x')$ as desired. By definition, this further implies $v_{a(x)}(x) \leq v_{a(x')}(x)$. Thus $w_{a(x)} \leq w_{a(x')}$, otherwise patient x has better utility at $a(x')$ than at $a(x)$, contradicting the stability of A. Therefore Claim 6.4 holds. □

Claim 6.4 shows that a is an ordered assignment function for any stable assignment $A = (a, w)$. Thus we immediately have the following claim.

CLAIM 6.5. *For any stable assignment $A = (a, w)$, there exists $x_0, \cdots, x_m$ with $0 = x_0 \leq x_1 \leq \cdots \leq x_{m-1} \leq x_m = 1$, such that for any $j \in [m]$ and $x \in (x_{j-1}, x_j)$, $a(x) = j$.*

Moreover, if A is optimal with respect to a, then $w_1 = 0$ and, for any $j \in \{2, \ldots, m\}$,

$$w_j = v_j(x_{j-1}) - v_{j-1}(x_{j-1}) + w_{j-1} = f_j(x_{j-1}) + w_{j-1} = \cdots = \sum_{k=1}^{j} f_k(x_{k-1}).$$

Claim 6.5 shows that, when A is stable and optimal with respect to a, for any $j > 1$, patient x_{j-1} is indifferent between hospitals $j-1$ and j. This claim is the counterpart of Lemmas 5.4, 5.6, and 5.7: to find the optimal stable assignment it suffices to focus on the choices of $x_0, \ldots, x_m$ and consider assignments that are tight with respect to them. The first part of Claim 6.5 follows directly from Claim 6.4, the first equality of the second part is similar to the analysis of Lemma 5.7, and the remaining of the second part is by induction. Thus we omit the detailed proof here. Now we are ready to prove Theorem 6.3.

PROOF OF THEOREM 6.3. Let $A = (a, w)$ be a stable assignment that is budget-feasible and optimal with respect to a, and $x_0, \ldots, x_m$ as specified in Claim 6.5: that is, for each $j \in [m]$, patients in (x_{j-1}, x_j) are assigned to hospital j. Since there is a continuous population of patients, it does not matter where patients $x_0, x_1, \ldots, x_m$ are assigned to. Moreover, the cost of A is

$$C(A) = \sum_{j \in [m]} c_j(x_j - x_{j-1}) \leq B.$$

Consider the randomized assignment $R = (p_1, \ldots, p_m)$ where $p_j = x_j - x_{j-1}$ for each $j \in [m]$. Note that $p_j \geq 0$ for each j and $\sum_{j \in [m]} p_j = \sum_{j \in [m]} x_j - x_{j-1} = x_m - x_0 = 1$. Thus R is well defined. Moreover, the cost of R is

$$C(R) = \sum_{j \in [m]} p_j c_j = \sum_{j \in [m]} c_j(x_j - x_{j-1}) = C(A) \leq B,$$

and R is budget-feasible.

Given this, we now show

$$SW(R) \geq SW(A). \tag{30}$$

When applied to $A = A^{opt}$, it implies Theorem 6.3.

To prove (30), notice that

$$\begin{aligned} SW(A) &= \int_0^1 [v_{a(x)}(x) - w_{a(x)}]dx = \sum_{j=1}^{m} \int_{x_{j-1}}^{x_j} [v_j(x) - w_j]dx \\ &= \sum_{j=1}^{m} \int_{x_{j-1}}^{x_j} \left[\sum_{k=1}^{j} f_k(x) - \sum_{k=1}^{j} f_k(x_{k-1})\right] dx = \sum_{j=1}^{m} \int_{x_{j-1}}^{x_j} \sum_{k=1}^{j} [f_k(x) - f_k(x_{k-1})]dx \\ &= \sum_{k=1}^{m} \sum_{j=k}^{m} \int_{x_{j-1}}^{x_j} [f_k(x) - f_k(x_{k-1})]dx = \sum_{k=1}^{m} \int_{x_{k-1}}^{1} [f_k(x) - f_k(x_{k-1})]dx, \end{aligned} \tag{31}$$

where the third equality is by the definition of $v_j(x)$ and Claim 6.5.

Moreover, by definition the social welfare of R is

$$SW(R) = \int_0^1 \sum_{j=1}^m p_j v_j(x)dx = \int_0^1 \sum_{j=1}^m (x_j - x_{j-1}) \sum_{k=1}^j f_k(x)dx = \sum_{k=1}^m \int_0^1 \left[\sum_{j=k}^m (x_j - x_{j-1})\right] f_k(x)dx$$
$$= \sum_{k=1}^m \int_0^1 (1 - x_{k-1}) f_k(x)dx. \quad (32)$$

We shall show

$$\int_0^1 (1 - x_{k-1}) f_k(x)dx \geq \int_{x_{k-1}}^1 [f_k(x) - f_k(x_{k-1})]dx \quad (33)$$

for every $k \in [m]$, which together with (31) and (32) implies (30). To do so, for any $k \in [m]$, consider the following function

$$g_k(y) = \int_0^1 (1-y) f_k(x)dx - \int_y^1 [f_k(x) - f_k(y)]dx = \int_0^1 (1-y) f_k(x)dx - \int_y^1 f_k(x)dx + (1-y) f_k(y)$$

for $y \in [0, 1]$. It suffices to show that $g_k(x_{k-1}) \geq 0$.

First, notice that

$$g_k(0) = \int_0^1 f_k(x)dx - \int_0^1 [f_k(x) - f_k(0)]dx = 0,$$

as $f_k(0) = 0$ by definition. Also,

$$g_k(1) = \int_0^1 0dx - \int_1^1 [f_k(x) - f_k(1)]dx = 0.$$

Moreover,

$$g'_k(y) = -\int_0^1 f_k(x)dx + f_k(y) - f_k(y) + (1 - y) f'_k(y) = (1 - y) f'_k(y) - \int_0^1 f_k(x)dx.$$

Because $(1 - y) f'_k(y)$ is non-increasing as required by Theorem 6.3, and because $\int_0^1 f_k(x)dx$ is a constant, we have that $g'_k(y)$ is non-increasing, that is, $g_k(y)$ is concave on $[0, 1]$. Since $g_k(0) = g_k(1) = 0$, we have $g_k(y) \geq 0$ for all $y \in [0, 1]$. Accordingly, $g_k(x_{k-1}) \geq 0$ and (33) holds. Thus (30) holds, and so does Theorem 6.3. □

COROLLARY 6.6. *If $f_j(x)$ is concave for every $j \in [m]$, then $SW(R^{opt}) \geq SW(A^{opt})$.*

Theorem 6.3 applies to many valuation functions that are not concave. For example, when the patients have proportional preferences, letting $v(x) = e^x - 1$, we have $(1 - x)v'(x) = (1 - x)e^x$, which is non-increasing on $[0, 1]$.

6.2. Optimal Lottery Schemes for Proportional Preferences

The structure of the optimal lottery schemes remains unknown for d-ordered preferences and is an important problem for future studies. Surprisingly, when the patients have proportional preferences, under the same condition as in Theorem 6.3, the optimal randomized assignment is in fact optimal among all lottery schemes. Recall that $v(x)$ is strictly increasing and $q_1 \leq \cdots \leq q_m$. We have the following.

THEOREM 6.7. *For proportional preferences, if $v(x)$ is almost concave then $SW(R^{opt}) = SW(L^{opt})$.*

To prove Theorem 6.7, we start by showing several properties of lottery schemes. Recall that given a lottery scheme L, for any patient $x \in [0,1]$, $\lambda^L(x) = (p_1^L(x), \ldots, p_m^L(x), w^L(x))$ denotes the lottery chosen by x.

LEMMA 6.8. *For any lottery scheme L, the function $\sum_{j\in[m]} p_j^L(x)q_j$ is non-decreasing.*

PROOF. Let $x, x' \in [0,1]$ be such that $x < x'$. By (29),

$$u(x, \lambda^L(x)) \geq u(x, \lambda^L(x')) \quad \text{and} \quad u(x', \lambda^L(x')) \geq u(x', \lambda^L(x)).$$

That is,

$$\left(\sum_{j\in[m]} p_j^L(x)q_j v(x)\right) - w^L(x) \geq \left(\sum_{j\in[m]} p_j^L(x')q_j v(x)\right) - w^L(x') \tag{34}$$

and

$$\left(\sum_{j\in[m]} p_j^L(x')q_j v(x')\right) - w^L(x') \geq \left(\sum_{j\in[m]} p_j^L(x)q_j v(x')\right) - w^L(x). \tag{35}$$

Adding the two inequalities side by side and rearranging the terms, we have

$$\left[\sum_{j\in[m]} p_j^L(x)q_j - \sum_{j\in[m]} p_j^L(x')q_j\right](v(x') - v(x)) \leq 0.$$

Since $v(x)$ is strictly increasing, we have $v(x) < v(x')$ and thus $\sum_{j\in[m]} p_j^L(x)q_j \leq \sum_{j\in[m]} p_j^L(x')q_j$. That is, the function $\sum_{j\in[m]} p_j^L(x)q_j$ is non-decreasing and Lemma 6.8 holds. □

Notice that the utility of patient x depends on x only indirectly through $v(x)$, thus $\lambda^L(x)$ can be written as a vector of functions on $v(x)$: $\lambda^L(v(x)) = (p_1^L(v(x)), \ldots, p_m^L(v(x)), w^L(v(x)))$. We have the following.

LEMMA 6.9. *For any lottery scheme L and any patient $x \in [0,1]$,*

$$u(x, \lambda^L(x)) = \int_0^{v(x)} \sum_{j\in[m]} q_j p_j^L(\hat{v})d\hat{v}.$$

PROOF. Similar to the proof of Lemma 6.8, by (29) and letting $x' = x + \Delta$, we have

$$u(x, \lambda^L(x)) \geq u(x, \lambda^L(x')) \quad \text{and} \quad u(x', \lambda^L(x')) \geq u(x', \lambda^L(x)).$$

That is,

$$v(x)\left[\sum_{j\in[m]} q_j \left(p_j^L(x+\Delta) - p_j^L(x)\right)\right] \leq w^L(x+\Delta) - w^L(x)$$

and

$$v(x+\Delta)\left[\sum_{j\in[m]} q_j \left(p_j^L(x+\Delta) - p_j^L(x)\right)\right] \geq w^L(x+\Delta) - w^L(x).$$

Combining the two inequalities and dividing each term by Δ, we have

$$v(x+\Delta)\left[\sum_{j\in[m]} q_j \frac{\left(p_j^L(x+\Delta)-p_j^L(x)\right)}{\Delta}\right] \geq \frac{w^L(x+\Delta)-w^L(x)}{\Delta}$$
$$\geq v(x)\left[\sum_{j\in[m]} q_j \frac{\left(p_j^L(x+\Delta)-p_j^L(x)\right)}{\Delta}\right].$$

Taking the limit as $\Delta \to 0$ and applying the definition of derivative on continuous functions, we have

$$v(x)\sum_{j\in[m]} q_j \cdot dp_j^L(x) = dw^L(x).$$

Accordingly,

$$\begin{aligned} du(x,\lambda^L(x)) &= \left(\sum_{j\in[m]} q_j \cdot d\left(p_j^L(x)v(x)\right)\right) - dw^L(x) \\ &= \sum_{j\in[m]} q_j p_j^L(x)\cdot dv(x) + v(x)\sum_{j\in[m]} q_j\cdot dp_j^L(x) - v(x)\sum_{j\in[m]} q_j\cdot dp_j^L(x) \\ &= \sum_{j\in[m]} q_j p_j^L(x)\cdot dv(x) = \sum_{j\in[m]} q_j p_j^L(v(x))v'(x)dx. \end{aligned}$$

Integrating the first and the last terms with respect to x and changing variables, we have

$$u(x,\lambda^L(x)) = \int_0^x du(\hat{x},\lambda^L(\hat{x})) = \int_0^x \sum_{j\in[m]} q_j p_j^L(v(\hat{x}))v'(\hat{x})d\hat{x} = \int_0^{v(x)} \sum_{j\in[m]} q_j p_j^L(\hat{v})d\hat{v}.$$

Thus Lemma 6.9 holds. □

Now we are ready to prove Theorem 6.7.

PROOF OF THEOREM 6.7. Since $SW(L^{opt}) \geq SW(R^{opt})$ by definition, it suffices to show

$$SW(R^{opt}) \geq SW(L^{opt}). \tag{36}$$

To do so, for any budget-feasible lottery scheme L, let $R = (p_1,\ldots,p_m)$ be the randomized assignment where for any $j \in [m]$,

$$p_j = \int_0^1 p_j^L(x)dx.$$

That is, each p_j is the average of $p_j^L(x)$ over $[0,1]$. It is easy to see $\sum_{j\in[m]} p_j = \int_0^1 \sum_{j\in[m]} p_j^L(x)dx \leq 1$, thus R is well defined. Also we have $C(R) = \sum_{j\in[m]} p_j c_j = \int_0^1 \sum_{j\in[m]} p_j^L(x)c_j dx = C(L) \leq B$, thus R is budget-feasible. Below we show

$$SW(R) \geq SW(L). \tag{37}$$

When applied to $L = L^{opt}$, it implies (36).

Since $v(x)$ is strictly increasing and twice differentiable, $v^{-1}(\hat{v})$ is well defined for any $\hat{v} \in [0, v(1)]$. Thus we have

$$SW(L) = \int_0^1 u(x, \lambda^L(x))dx = \int_0^1 \int_0^{v(x)} \sum_{j\in[m]} q_j p_j^L(\hat{v}) d\hat{v} dx = \int_0^{v(1)} \left(\sum_{j\in[m]} q_j p_j^L(\hat{v}) \int_{v^{-1}(\hat{v})}^1 dx \right) d\hat{v}$$
$$= \int_0^{v(1)} \sum_{j\in[m]} q_j p_j^L(\hat{v})(1 - v^{-1}(\hat{v}))d\hat{v} = \int_0^1 \sum_{j\in[m]} q_j p_j^L(x)(1-x)v'(x)dx, \tag{38}$$

where the second equality is by Lemma 6.9, the last is by taking $\hat{v} = v(x)$, and all others are by definition or basic calculus. Moreover, letting $p = \sum_{j\in[m]} q_j p_j$, we have

$$SW(R) = \int_0^1 pv(x)dx = \int_0^1 \int_0^{v(x)} p d\hat{v} dx = \int_0^{v(1)} \int_{v^{-1}(\hat{v})}^1 p dx d\hat{v} = \int_0^{v(1)} p(1 - v^{-1}(\hat{v}))d\hat{v}$$
$$= \int_0^1 p(1-x)v'(x)dx. \tag{39}$$

By Lemma 6.8, $\sum_{j\in[m]} q_j p_j^L(x)$ is non-decreasing. Thus

$$\begin{aligned} \sum_{j\in[m]} q_j p_j^L(0) &= \int_0^1 \sum_{j\in[m]} q_j p_j^L(0)dx \le \int_0^1 \sum_{j\in[m]} q_j p_j^L(x)dx \\ &= p \le \int_0^1 \sum_{j\in[m]} q_j p_j^L(1)dx = \sum_{j\in[m]} q_j p_j^L(1), \end{aligned}$$

and there exists $x_p \in [0, 1]$ such that

$$\sum_{j\in[m]} q_j p_j^L(x) \le p \ \forall x \in [0, x_p) \quad \text{and} \quad \sum_{j\in[m]} q_j p_j^L(x) \ge p \ \forall x \in [x_p, 1].$$

Following (38) and (39) we have

$$\begin{aligned} SW(R) - SW(L) &= \int_0^1 \left(p - \sum_{j\in[m]} q_j p_j^L(x) \right) (1-x)v'(x)dx \\ &= \int_0^{x_p} \left(p - \sum_{j\in[m]} q_j p_j^L(x) \right) (1-x)v'(x)dx \\ &\quad + \int_{x_p}^1 \left(p - \sum_{j\in[m]} q_j p_j^L(x) \right) (1-x)v'(x)dx. \end{aligned} \tag{40}$$

The value of $\sum_{j\in[m]} q_j p_j^L(x_p)$ does not affect the value of the integration, and without loss of generality we assume it equals p.

For any $x \in [0, x_p]$, since $(1-x)v'(x)$ is non-increasing and $v'(x) > 0$, we have

$$(1-x)v'(x) \ge (1-x_p)v'(x_p) \ge 0.$$

Moreover, for any such x, since $p - \sum_{j\in[m]} q_j p_j^L(x) \geq 0$, we have

$$\left(p - \sum_{j\in[m]} q_j p_j^L(x)\right)(1-x)v'(x) \geq \left(p - \sum_{j\in[m]} q_j p_j^L(x)\right)(1-x_p)v'(x_p). \qquad (41)$$

Similarly, for any $x \in [x_p, 1]$ we have $0 \leq (1-x)v'(x) \leq (1-x_p)v'(x_p)$ and $p - \sum_{j\in[m]} q_j p_j^L(x) \leq 0$, and thus

$$\left(p - \sum_{j\in[m]} q_j p_j^L(x)\right)(1-x)v'(x) \geq \left(p - \sum_{j\in[m]} q_j p_j^L(x)\right)(1-x_p)v'(x_p). \qquad (42)$$

Combining (40) with (41) and (42), we have

$$\begin{aligned}
& SW(R) - SW(L) \\
\geq\ & \int_0^{x_p} \left(p - \sum_{j\in[m]} q_j p_j^L(x)\right)(1-x_p)v'(x_p)dx + \int_{x_p}^1 \left(p - \sum_{j\in[m]} q_j p_j^L(x)\right)(1-x_p)v'(x_p)dx \\
=\ & (1-x_p)v'(x_p)\int_0^1 \left(p - \sum_{j\in[m]} q_j p_j^L(x)\right)dx = (1-x_p)v'(x_p)\left[p - \int_0^1 \sum_{j\in[m]} q_j p_j^L(x)dx\right] \\
=\ & (1-x_p)v'(x_p)(p-p) = 0,
\end{aligned}$$

implying (37). Thus Theorem 6.7 holds. □

COROLLARY 6.10. *For any concave valuation function $v(x)$, $SW(R^{opt}) = SW(L^{opt})$.*

Theorem 6.7 also applies to many valuation functions that are not concave, such as $v(x) = e^x - 1$ as we have seen.

6.3. Discussions and Important Properties of the Optimal Randomized Assignment

From the Continuous Model to the Discrete Model. When the set of patients is not $[0,1]$ but $\{1,\ldots,n\}$ as in Sections 4 and 5, the valuation functions v_j and difference functions f_j with $j \in [m]$ are defined in the same way over $[n]$ for d-ordered preferences, so is the valuation function v for proportional preferences. Given a discrete function f over a domain $D \subseteq \mathbb{Z}^+$, for any $x \in D$, the *difference* of f at x is $\Delta_x f(x) \triangleq f(x+1) - f(x)$. Again, we add a dummy patient 0 such that $v_j(0) = 0$ for any $j \in [m]$. Replacing derivatives by differences, the condition in Theorem 6.3 becomes that, for each hospital $j \in [m]$, the sequence $(n-i)\Delta_i f_j(i)$ with $i \in \{0,1,\ldots,n-1\}$ is non-increasing; and that in Theorem 6.7 becomes that the sequence $(n-i)\Delta_i v(i)$ with $i \in \{0,1,\ldots,n-1\}$ is non-increasing. Further replacing integrations by sums in the analysis, it is easy to see that the discrete versions of Theorems 6.3 and 6.7 hold —details have been omitted.

Theorems 6.3 and 6.7 in Terms of Monotone Hazard Rate. Interestingly, although we approached our results solely from the aspect of lottery schemes, we have discovered afterward that the condition of almost concavity in Theorems 6.3 and 6.7 have a natural interpretation from another aspect. Consider an assignment problem where there is a single patient and $[m]$ hospitals. The planner's budget is lower than the most expensive hospital, thus the patient cannot simply be assigned to his favorite hospital with probability 1. There is a distribution D from which his values for the hospitals are drawn: first draw his *type* x uniformly at random from $[0,1]$, then compute $f_j(x)$ for each $j \in [m]$ and set his value for hospital j to be $v_j(x) = \sum_{k=1}^{j} f_j(x)$. Arbitrarily fix a hospital j and consider the random variable $Y_j = f_j(X)$, with X distributed according to $U[0,1]$. It is easy to see that for any $y_0 \in f_j([0,1])$ and $x_0 = f_j^{-1}(y_0)$, the cumulative distribution function of Y_j at y_0 is

$$F_j(y_0) = \Pr[Y_j \le y_0] = \Pr[f_j^{-1}(Y_j) \le x_0] = \Pr[X \le x_0] = x_0 = f_j^{-1}(y_0),$$

and the probability density function at y_0 is

$$d_j(y_0) = F_j'(y_0) = \frac{1}{f_j'(x_0)}.$$

Accordingly, the formula in Theorem 6.3 becomes

$$(1 - x_0) f_j'(x_0) = \frac{1 - F_j(y_0)}{d_j(y_0)} = \frac{1}{h_j(y_0)},$$

where $h_j(y) \triangleq \frac{d_j(y)}{1-F_j(y)}$ is the *hazard rate* of Y_j. Since $f_j(x)$ is strictly increasing, $(1 - x)f_j'(x)$ is non-increasing *if and only if* $h_j(y)$ is non-decreasing: that is, Y_j has *monotone hazard rate* (MHR). Thus we immediately have the following.

COROLLARY 6.11. *For d-ordered preferences, if Y_j has MHR for each $j \in [m]$ then $SW(R^{opt}) \ge SW(A^{opt})$.*

Similarly, for proportional preferences, consider the distribution D induced by the valuation function $v(x)$ and we have the following.

COROLLARY 6.12. *For proportional preferences, if D has MHR then $SW(R^{opt}) = SW(L^{opt})$.*

Computation of the Optimal Randomized Assignment. In general Provision-after-Wait problems, there may not be an efficient algorithm for computing an optimal lottery scheme. However, the optimal randomized assignment $R^{opt} = (p_1, \dots, p_m)$ is simply defined by the following linear program.

$$\begin{aligned} &\max_{p_1,\dots,p_m} \sum_{j\in[m]} p_j \int_0^1 v_j(x)dx \\ \text{s.t. } & p_j \ge 0 \;\forall j \in [m], \\ &\sum_{j\in[m]} p_j \le 1, \\ &\sum_{j\in[m]} p_j c_j \le B. \end{aligned}$$

Note the computability of R^{opt} and $SW(R^{opt})$ depends completely on the description of v_j's. If for every $j \in [m]$, $\int_0^1 v_j(x)dx$ has a closed form and can be computed in polynomial time, then R^{opt} and $SW(R^{opt})$ can be computed in polynomial time, since each $\int_0^1 v_j(x)dx$ is a constant in the linear program. Otherwise, by computing each $\int_0^1 v_j(x)dx$ numerically, R^{opt} and $SW(R^{opt})$ can also be computed numerically.

Ex-Post Budget-Feasibility. How to implement lotteries so that the desired constraints are satisfied ex-post is an important research topic in the Economics literature; see, e.g., [Budish et al. 2013]. In the Provision-after-Wait problem, a lottery scheme in general only satisfies the budget constraint in expectation, and it is possible that under some realization of the lotteries the total cost is much higher than the budget. However, given a randomized assignment $R = (p_1, \dots, p_m)$, the planner can first choose a permutation of the patients uniformly at random, and then assign the first p_1 *fraction* of them to hospital 1, the next p_2 fraction to hospital 2, etc. By doing so, each patient is assigned to the hospitals according to the correct distribution $(p_1, \dots, p_m)$, thus the expected social welfare equals $SW(R)$. While in any realized assignment the total cost is $\sum_{j\in[m]} p_j c_j$, exactly the expected cost of the randomized assignment, and thus the budget constraint is satisfied with probability 1.

Advantage in Generating Social Welfare. When Theorem 6.3 or 6.7 applies, not only the social welfare of the optimal randomized assignment is no less that of the optimal stable assignment, but the ratio between the two can be arbitrarily large, since in the latter a lot of social welfare may be burnt by letting the patients wait. As an example, consider the case of proportional preferences where $v(x) = v_0$ is a positive constant, $q_1 = \epsilon \ll 1$, $1 \ll q_2 < \cdots < q_m$, $B \gg 1$, $c_1 = 1$, $c_2 = \cdots = c_m = \frac{B-\epsilon}{1-\epsilon}$. It is easy to see that one particular optimal stable assignment is to assign all patients to hospital 1 with waiting time 0, where the social welfare is $q_1 \int_0^1 v(x)dx = \epsilon v_0$: assigning some patients to better hospitals won't help, since $v(x)$ is a constant and all patients must have the same utility. However, there is a randomized assignment that assigns each patient to hospital m with probability $1-\epsilon$ and to hospital 1 with probability ϵ, resulting in total cost $(1-\epsilon)c_m + \epsilon c_1 = B$ and social welfare $((1-\epsilon)q_m + \epsilon q_1) \int_0^1 v(x)dx = ((1-\epsilon)q_m + \epsilon^2)v_0 \geq (1-\epsilon)v_0 \gg \epsilon v_0$. To make $v(x)$ strictly increasing, just take $v(x) = v_0 + \alpha x$ with some arbitrarily small $\alpha > 0$: the analysis is essentially the same as when $v(x)$ is a constant.

An Example Where Stable Assignments Can Do Better. Finally, we provide a simple example with proportional preferences where the valuation function v is not almost concave and stable assignments are better than randomized assignments. Consider two hospitals 0 and 1, with $c_0 = q_0 = 0$, $c_1 > 0$ and $q_1 = 1$. The patients have proportional preferences with $v(x) = e^{2x} - 1$ and the budget is $B \in [0, c_1]$.

Note that $(1-x)v'(x) = 2(1-x)e^{2x}$, which is increasing when $x \leq 1/2$ and decreasing otherwise, so Theorem 6.3 or 6.7 does not apply here. It is easy to see that the optimal randomized assignment is $R^{opt} = (p_0, p_1)$ where $p_1 = \frac{B}{c_1}$ and $p_0 = 1 - p_1$. We have $C(R^{opt}) = B$ and

$$SW(R^{opt}) = \frac{B}{c_1} \int_0^1 (e^{2x} - 1)dx = \frac{B(e^2-3)}{2c_1}.$$

By Claim 6.5, for any stable assignment $A = (a, w)$ that is budget-feasible with a ordered, there exists $x_1 \in [0,1]$ such that patients in $(0, x_1)$ are assigned to hospital 0, patients in $(x_1, 1)$ are assigned to hospital 1, $w_0 = 0$ and $w_1 = e^{2x_1} - 1$. Accordingly, $C(A) = c_1(1-x_1) \leq B$ and

$$SW(A) = \int_{x_1}^1 v(x) - w_1 dx = \int_{x_1}^1 e^{2x} - e^{2x_1} dx = \frac{e^2 + e^{2x_1}(2x_1 - 3)}{2}.$$

Therefore the optimal stable assignment A^{opt} is such that $x_1 = 1 - \frac{B}{c_1}$. Letting $r = \frac{B}{c_1}$, we have

$$SW(R^{opt}) = \frac{(e^2-3)r}{2} \quad \text{and} \quad SW(A^{opt}) = \frac{e^2 - e^{2(1-r)}(1+2r)}{2}.$$

Thus there exists $r_0 \approx 0.8$ such that $SW(R^{opt}) \geq SW(A^{opt})$ if $r \leq r_0$ and $SW(R^{opt}) < SW(A^{opt})$ otherwise. That is, if the budget is enough to serve 80 percent of the patients at hospital 1 then the optimal stable assignment has better social welfare, otherwise the optimal randomized assignment does better. Figure 1 shows the difference between the two for $r \in [0, 1]$. In this example, the optimal stable assignment only has a small advantage when $r > r_0$: $\max_r SW(A^{opt})/SW(R^{opt}) \approx 1.005$, which occurs at $r \approx 0.9$.

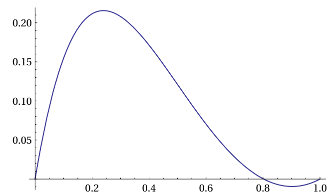

Fig. 1. $SW(R^{opt}) - SW(A^{opt})$ as a function of $r = \frac{B}{c_1}$.

7. EXTENSIONS AND OPEN PROBLEMS

Our results naturally extends to the more general setting where there are multiple medical services and each hospital may have a different cost for each service.[9] In this case, rather than deciding how much each hospital gets paid out of the total budget, the planner decides how much each hospital gets paid for each service. Different patients may require different services and one patient only wants one of them. The patients may still have common preferences with respect to each service, but the ranks of the hospitals may be different for different services. Indeed, in real life hospitals may have service-dependent reputations. Also, in reality patients requiring different services may face different waiting times at the same hospital. Thus we allow each hospital to have a waiting time for each service. When the patients are d-ordered with respect to each service, our characterization for optimal stable assignments applies to each service separately and we can construct an FPTAS for computing a *globally budget-feasible* stable assignment that approximately maximizes the social welfare. Moreover, a lottery scheme can also be decomposed into separate lottery schemes for different services. For each service, fixing the budget given to this service, Theorems 6.3 and 6.7 continue to hold. However, the globally budget-feasible randomized assignment with maximum social welfare is computed by a linear program with an additional constraint: that is, the sum of the budget for each service is no more than the total budget B.

The structure of the optimal stable assignment when the patients are not d-ordered remains unknown, and an important open problem for future studies is to understand the general class of common preferences. That is, when the patients are not d-ordered but still have the same ranking for the hospitals, is the problem strongly NP-hard or is there an FPTAS? A good starting point would be patients' preferences that are "piece-wise d-ordered": for example, there exists a hospital j such that the patients are d-ordered both on hospitals $\{1, \ldots, j\}$ and on hospitals $\{j, \ldots, m\}$, but the orders are different for the two pieces. Although the two pieces in the example can be considered as two different services, here each patient may receive either service and the problem cannot be decomposed with respect to the services. Our conjecture is that the problem is still hard even in this setting and it would be interesting to see a clear answer.

When lottery is used as a rationing tool, our results show that even lottery schemes as simple as randomized assignments can effectively avoid welfare-burning. If the planner is allowed to ask the patients to enter lotteries, then it is a good idea to combine randomness with waiting times so as to improve social welfare. In particular, it

[9]We thank an anonymous reviewer for motivating us to consider this extension.

would be great to characterize the structure of optimal lottery schemes for both d-ordered preferences and proportional preferences, although the problem may be much easier for the latter than for the former.

On a related but different front, note that part of the difficulties in using waiting times as a rationing tool comes from the fact that the planner's budget and the patients' waiting times are two unexchangeable "currencies". We would be interested in healthcare provision problems where the planner can distribute subsidies to the patients and where the subsidies may be a bridge connecting the budget and the waiting times. Finally, problems with prioritized waiting times and problems when each patient wants a set of medical services are all interesting extensions of our model.

REFERENCES

Atila Abdulkadiroğlu, Parag A. Pathak, and Alvin E. Roth. 2005. The New York City High School Match. *American Economic Review* 95, 2 (May 2005), 364–367.

Saeed Alaei, Azarakhsh Malekian, and Aravind Srinivasan. 2014. On Random Sampling Auctions for Digital Goods. *ACM Trans. Econ. Comput.* 2, 3 (2014), 11:1–11:19.

Itai Ashlagi and Peng Shi. 2014. Improving Community Cohesion in School Choice via Correlated-Lottery Implementation. *Oper. Res.* 62, 6 (Dec. 2014), 1247–1264.

Susan Athey and Glenn Ellison. 2009. *Position Auctions with Consumer Search*. NBER Working Papers 15253. National Bureau of Economic Research, Inc.

Anna Bogomolnaia and Herve Moulin. 2001. A New Solution to the Random Assignment Problem. *Journal of Economic Theory* 100, 2 (October 2001), 295–328.

M. Braverman, J. Chen, and S. Kannan. 2016. Optimal Provision-After-Wait in Healthcare. *Mathematics of Operations Research* 41, 1 (2016), 352–376.

Patrick Briest. 2008. Uniform Budgets and the Envy-Free Pricing Problem. In *Automata, Languages and Programming: 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part I*. Springer Berlin Heidelberg, Berlin, Heidelberg, 808–819.

Patrick Briest, Shuchi Chawla, Robert Kleinberg, and S. Matthew Weinberg. 2010. Pricing Randomized Allocations. In *Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '10)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 585–597.

Eric Budish, Yeon-Koo Che, Fuhito Kojima, and Paul Milgrom. 2013. Designing Random Allocation Mechanisms: Theory and Applications. *American Economic Review* 103, 2 (2013), 585–623.

H. Chan and J. Chen. 2016. Provision-After-Wait with Common Preferences. In *15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'16),* to appear.

Shuchi Chawla, Jason D. Hartline, and Robert Kleinberg. 2007. Algorithmic Pricing via Virtual Valuations. In *Proceedings of the 8th ACM Conference on Electronic Commerce (EC '07)*. ACM, New York, NY, USA, 243–251.

Shuchi Chawla, David L. Malec, and Balasubramanian Sivan. 2010. The Power of Randomness in Bayesian Optimal Mechanism Design. In *Proceedings of the 11th ACM Conference on Electronic Commerce (EC '10)*. ACM, New York, NY, USA, 149–158.

Xi Chen, Ilias Diakonikolas, Dimitris Paparas, Xiaorui Sun, and Mihalis Yannakakis. 2014. The Complexity of Optimal Multidimensional Pricing. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '14)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1319–1328.

Julie Berry Cullen, Brian A Jacob, and Steven Levitt. 2006. The Effect of School Choice on Participants: Evidence from Randomized Lotteries. *Econometrica* 74, 5 (2006), 1191–1230.

Diane Dawson, Hugh Gravelle, Rowena Jacobs, Stephen Martin, and Peter C. Smith. 2007. The effects of expanding patient choice of provider on waiting times: evidence from a policy experiment. *Health Economics* 16, 2 (2007), 113–128.

David J. Deming, Justine S. Hastings, Thomas J. Kane, and Douglas O. Staiger. 2014. School Choice, School Quality and Postsecodary Attainment. *The American Economic Review* 104, 3 (2014), 991–1013.

Nikhil R. Devanur, Bach Q. Ha, and Jason D. Hartline. 2013. Prior-free Auctions for Budgeted Agents. In *Proceedings of the Fourteenth ACM Conference on Electronic Commerce (EC '13)*. ACM, New York, NY, USA, 287–304.

Stefan Felder. 2008. To wait or to pay for medical treatment? Restraining ex-post moral hazard in health insurance. *Journal of Health Economics* 27, 6 (2008), 1418–1422.

H. Gravelle and L. Siciliani. 2008a. Is waiting-time prioritisation welfare improving? *Health Economics* 17, 2 (2008), 167–184.

Hugh Gravelle and Luigi Siciliani. 2008b. Optimal quality, waits and charges in health insurance. *Journal of Health Economics* 27, 3 (2008), 663–674.

Hugh Gravelle and Luigi Siciliani. 2009. Third Degree Waiting Time Discrimination: Optimal Allocation of a Public Sector Health Care Treatment Under Rationing by Waiting. *Health Economics* 18, 8 (2009), 977–986.

Venkatesan Guruswami, Jason D. Hartline, Anna R. Karlin, David Kempe, Claire Kenyon, and Frank McSherry. 2005. On Profit-maximizing Envy-free Pricing. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '05)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1164–1173.

Bach Q. Ha and Jason D. Hartline. 2013. Mechanism Design via Consensus Estimates, Cross Checking, and Profit Extraction. *ACM Trans. Econ. Comput.* 1, 2 (May 2013), 8:1–8:15.

J.D. Hartline and T. Roughgarden. 2008. Optimal mechanism design and money burning. In *Proceedings of the 40th annual ACM Symposium on Theory of Computing*. ACM, 75–84.

Jason Hartline and Qiqi Yan. 2011. Envy, Truth, and Profit. In *Proceedings of the 12th ACM Conference on Electronic Commerce (EC '11)*. ACM, New York, NY, USA, 243–252.

A. S. Kelso and V. P Crawford. 1982. Job Matching, Coalition Formation, and Gross Substitutes. *Econometrica* 50, 6 (November 1982), 1483–1504.

Stavros G. Kolliopoulos and George Steiner. 2007. Partially Ordered Knapsack and Applications to Scheduling. *Discrete Appl. Math.* 155, 8 (April 2007), 889–897.

Victor Lavy. 2010. Effects of Free Choice Among Public Schools. *The Review of Economic Studies* 77, 3 (2010), 1164–1191.

Alejandro M. Manelli and Daniel R. Vincent. 2006. Bundling as an optimal selling mechanism for a multiple-good monopolist. *Journal of Economic Theory* 127, 1 (March 2006), 1–35.

Silvano Martello and Paolo Toth. 1990. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, Inc.

L. Siciliani and J. Hurst. 2005. Tackling excessive waiting times for elective surgery: a comparative analysis of policies in 12 OECD countries. *Health policy* 72, 2 (2005), 201–215.

John Thanassoulis. 2004. Haggling over substitutes. *Journal of Economic Theory* 117, 2 (August 2004), 217–245.

Hal R. Varian. 2007. Position auctions. *International Journal of Industrial Organization* 25, 6 (December 2007), 1163–1178.